**The Thesis Committee for Francisco Ochoa**
**Certifies that this is the approved version of the following Thesis:**

# Vegetation indices and the calibration of transpiration models in the American Southwest

**APPROVED BY**
**SUPERVISING COMMITTEE:**

William Doolittle, Supervisor

Thoralf Meyer

Ashely Michelle Matheny

# Vegetation indices and the calibration of transpiration models in the American Southwest

**by**

**Francisco Ochoa**

**Thesis**

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

**Master of Arts**

**The University of Texas at Austin**

**May 2020**

# Dedication

To my mother and father, Nora and Francisco, who showed me that the Southwest, our home, is filled with hidden gems and gardens awaiting their rediscovery.

# Acknowledgements

The findings of this project would not have been possible without the academic and financial support from the Department of Geography and Environment at The University of Texas at Austin. I would like to thank the support from Dr. Kelley Crews and the Digital Landscape Laboratory, as well as Dr. William Doolittle for their guidance throughout the research process. The contributions and support from Dr. Ashely Matheny, from Jackson School of Geosciecnes, and Dr. Thoralf Meyer made this project possible.

## Abstract

## Vegetation indices and the calibration of transpiration models in the American Southwest

Francisco Ochoa, GRG

The University of Texas at Austin, 2020

Supervisor: William Doolittle


Land management decisions should incorporate reliable estimates of energy and water available within a managed ecosystem or watershed. Studies in Central Texas savannas and in Southern New Mexico's open-shrublands have demonstrated that evapotranspiration accounts for 60% to 90% of incoming precipitation within the same watershed. Models of evapotranspiration from various governmental agencies have been incorporated into different water conservation plans and policies in multiple southwestern states which in turn supported land and wildlife management decisions.

Using the International Geosphere-Biosphere Programme vegetation classification system, carbon uptake was analyzed at a diurnal scale to assess the impact of timing (of space as well as ground derived measurements) on the calibration of transpiration models based on vegetation indices that incorporate photosynthetic active radiation (0.38 – 0.70 μm). Beer-Lambert's Law was used to model the non-linear and linear relationships that exists between crop coefficients derived from eddy covariance systems and MODIS based measurements of various vegetation indices at different temporal scales in woody savannas, open-shrublands, and grasslands of the American southwest. Using high temporal resolution ground-based measurements of the normalized vegetation index along a southern California climate gradient, we determined the range and peak values and compared them to values derived from MODIS' *Terra* and *Aqua* satellites at their respective times of overpass in the study area. Results indicate that temporal differences [on a 24 hour basis] need to be considered during the calibration of estimates involving vegetation indices and transpiration modeling.

# Table of Contents

# List of Tables

# List of Figures

# 1. INTRODUCTION

Over the past centuries, grasslands and savannas around the world have been subject to a phenomena referred to as woody encroachment, in which woody plants replace or outcompete native grasses (Ratajczak, Nippert, and Collins 2012). During the mid-1900s, several efforts were made by local, state, and federal governments to manage this phenomena with the overall goal of increasing carrying capacities of cattle ranches across the semi-arid regions of the American Southwest (Sayre 2017). Studies have indicated that woody encroachment in savannas has been linked to modifications of the $CO_2$ exchange between the ground and atmosphere (J. Wang et al. 2018) as well as eco-hydrological functions such as streamflow, runoff, and recharge (Huxman et al. 2005). The work of D'Odorico et al. (2010) demonstrated how changes to surface characteristics (e.g. surface albedo) are able to create a positive feedback loop that amplifies the survival and establishment of *Larrea tridentata* through interactions with surface temperature and hydrology. As temperature and atmospheric levels of $CO_2$ continue to rise across the world, vegetation models suggest that woody plants will be the dominant species across multiple biomes (Briske 2000) which, in turn, will present new challenges in the management of water across the landscape scale.

The $H_2O$ cycle is commonly defined as a general system that describes the interactions of $H_2O$ moving between the lithosphere, biosphere, and atmosphere in different physical states (Chapin 2002). Positive values of latent heat flux ($\lambda LE$) can be associated with the movement of $H_2O$ into the atmosphere from the land surface through means of evaporation from the soil, open water bodies, water intercepted by leaves, and transpiration from vegetation; when $\lambda LE$ is negative, water from the atmosphere is moving to the land surface through condensation (Guyot 1998). A full understanding of how water moves throughout ecosystems also allows for proper management decisions in regards to land use, agriculture, wildlife population control, conservation, or mitigation of invasive species.

$\lambda LE$ is considered to be a major source of $H_2O$ depletion in arid lands. In the Jornada Experimental Range (JER) in New Mexico, a semiarid shrubland, $\lambda LE$ has been observed to account for 72 % - 95 % of incoming precipitation along piedmont slopes (Schreiner-McGraw and Vivoni 2017; Gutschick and Snyder 2006). In Freeman Ranch, a woody savanna in Central Texas, $\lambda LE$ has been observed to account for approximately 60 % of incoming precipitation (Kjelgaard et al. 2008). The Luck Hills site of the Walnut Experimental Watershed situated in Arizona reported that woody shrubs account for 70 % of $\lambda LE$ during the growing season (Scott et al. 2006). With increasing water scarcity in the American Southwest and other parts of the world, $H_2O$ and land managed on the basis of science to inform responsible management decisions.

Several efforts in the field of land management have been made in order to minimize this major depletion of the $H_2O$ cycle caused by woody encroachment. In the JER, Gee et al. (1994) attributed 50 % increase in groundwater recharge from annual precipitation once *L. tridentata* was cleared along a piedmont slope. In the Edwards Aquifer of Central Texas, multiple efforts have been made to understand how removing *Juniperus ashei* could increase the recharge rates of the aquifer. Results demonstrated that *J. ashei* is able to obtain groundwater through its deep penetrating roots during the dry periods as opposed to its shallow roots during the rainy season (McCole and Stern 2007). The work of Dugas, Hicks, and Wright (1998) also demonstrated that by removing *J. ashei*, daily values of $\lambda LE$ decreased by 7 %.

Congress passed the Saltcedar and Russian Olive Control Act in 2006 which granted $80 million dollars in funding for research and demonstration projects to remove

Saltcedar (*Tamarix spp.*) in the reservoirs and rivers of American Southwest (Doughty 2019). Along the Pecos River in Texas and New Mexico, \$2.2 - \$2.7 million dollars of federal, state, and local funds were spent removing *Tamarisk* with the overall goal to increase surface flow to irrigation districts by reducing $\lambda LE$ from invasive species (Sher et al. 2013). Improved measurements of $\lambda LE$ can help farmers, ranchers, and city planners estimate water budgets at both local and regional scales to increase crop yield (Ko and Piccinni 2009), increase unconfined aquifer recharge in agricultural communities through irrigation (Ochoa et al. 2013), and mitigate the risks associated with increasing water shortages and droughts.

Unlike precipitation, which can be measured with high precision and accuracy using a rain tipping bucket, $\lambda LE$ remains invisible to human eyesight. $\lambda LE$ can be directly measured at hyper temporal scales using various techniques such as the Bowen Ratio or an Eddy Covariance system (EC). EC uses the principle of the surface energy balance (equation 1) and atmospheric eddies to model incoming/outgoing energy fluxes, where $Rn$ is net radiation (W/m$^2$), $G$ is the ground heat flux (W/m$^2$), and $H$ is the sensible heat flux (W/m$^2$). EC instrumentation is expensive with costs approaching \$50,000 - \$ 75,000 USD for one station and coverage is spatially limited. Ameriflux is an EC data portal operated by the Department of Energy (DOE) Office of Science that makes these data free for a diverse range of ecosystems across the world. As of January 2020, there are 474 EC sites registered through Ameriflux.

$$\lambda LE = Rn - G - H \tag{1}$$

Throughout the last two decades, there have been various efforts led by researchers and governmental agencies to quantify $\lambda LE$ through the use of passive-reflectance and passive-emission remote sensing platforms in order to fill in the knowledge for this major depletion in the water system across the landscape and global scale (Vinukollu et al. 2011).

Remote sensing platforms in a Sun-synchronous or Geostationary orbit as well as ground mounted multi-spectral cameras on the terrestrial surface allow users to see different regions of the electromagntetic spectrum and how they interact with its surrounding physical environment. Through these observations, scientists have been able to develop relationships with ground-turhting to determine estimated readings of, spectral emissivity, surface albedo, vegetation indices, and land surface temperature (Stofan et al. 2007; Mustard 2017; Jimenez-Munoz et al. 2014). The orbital mechanics of sun-synchrounous satellites allow for two daily scene aquistions using MODIS or once every 16 days when using Landsat. This orbital track might present a problem when measuring a diurnal process, such as photosynthesis.

Photosyntheis has been demonstrated to fluxate at a diurnal scale which in turns affects the transpiration rates of vegetation through stomatal conductance (Farquhar and Sharkey 2003). When measuring $\lambda LE$ in arid and semi-arid lands, which cover 30 % - 45 % of the terrestrial surface (Hastings, Oechel, and Muhlia-Melo 2005), estimates have to account for photosynthetic pathways of the dominant vegetation. Photosynthesis is the biochemical process in which photosynthetic active radiation (PAR) (400 – 700 nm) is absorbed by living plants to produce carbohydrates ($CH_2O$) such as sugars and starches from carbon dioxide ($CO_2$) and $H_2O$ (Lambers, Chapin III, and Pons 2008; Bonan 2015). Photosynthesis can occur through various carbon chain pathways after PAR is absorbed causing the excitation of Chlorophyll. The pathways for photosynthesis in plant cells are $C_3$, $C_4$, and Crassulacean acid metabolism (CAM). The majority of flowering and woody plant species follow the $C_3$ pathway and only 3% of species, mostly grasses in the family Poaceae, use the $C_4$ pathway (Kellogg 2013). The geographic distribution of $C_3$ and $C_4$ grasslands and shrublands was mapped across the United States through climatic data and vegetation' survey's across 73 sites; the southwestern states of Texas and New Mexico are covered by an abundance of $C_4$ grasslands and shrublands (Figure 1) (Paruelo and Lauenroth 1996).

As plant becomes water stressed, due to dry soil of high atmospheric vapor pressure deficit (VPD), stomata begin to close to limit water loss (Lambers, Chapin III, and Pons 2008; Bonan 2015). With water limitations in the atmosphere or soil, the main differences and efficiencies between $C_3$ and $C_4$ can be observed. Both pathways absorb $CO_2$ through the stomata but the primary difference lies in the location where carbon fixation takes place. In the $C_3$ pathway, the Calvin Cycle occurs in the mesophyll cells. While in $C_4$, the initial carbon fixation takes place withinin the mesophyll and the Calvin Cycle occurs in the bundle sheath cell (Bonan 2015; C. Wang et al. 2012). This separation of cells causes $C_4$ plans to be more efficient at carbon uptake, since they are able to uptake $CO_2$ at a point during the day which conditions are favorable (a low VPD), close the stomata as temperature increases, and store the $CO_2$ within the mesophyll cells, which allows the plant to have a supply of $CO_2$ while only opening stomata minimally. Another key difference is that through the $C_3$ pathway, $CO_2$ is lost through photorespiration, while in $C_4$ very little or none takes place. Photorespiration is the process of oxygenation of Ribulose 1, 5-bisphosphate (RuBP) by Ribulose-1, 5-bisphosphate carboxylase/oxygenase (RuBisCO), which can reduce total carbon uptake by quantities greater than 40% (Sage, Wedin, and Li 1999).

A photosynthetically active leaf has a an absorption peak of chlorophyll centered at around 0.40 μm and 0.72 μm while $H_2O$ has two absorptions peaks centered at 1.45 μm and 1.95 μm (Tempfli et al. 2009; Gurevitch, Scheiner, and Fox 2002; Huete et al. 1997; Okin et al. 2001). These plant photosynthetic functionalities affect the depth of absorptions caused by chlorophyll, mesophyll, and $H_2O$ which are used to calculate vegetation indices

(Figure 2). The more water that is available within the leaf, the higher is the absorption rate of $H_2O$.

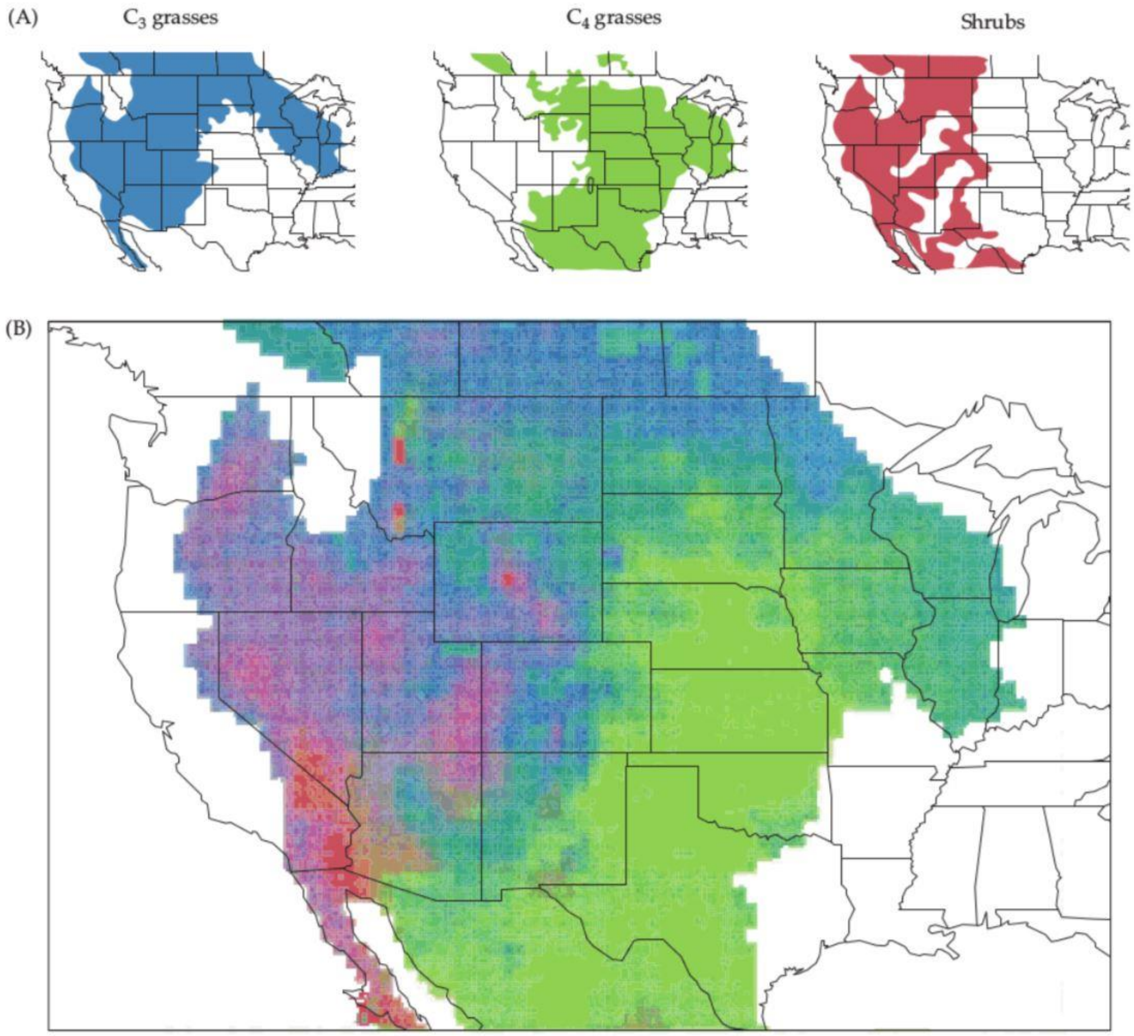


Figure 1: Distribution of $C_3$ and $C_4$ grasslands and shrublands in parts of the Continental United States. Fiugre taken from Gurevitch, Scheiner, and Fox (2002) and based on model produced by Paruelo and Lauenroth (1996).

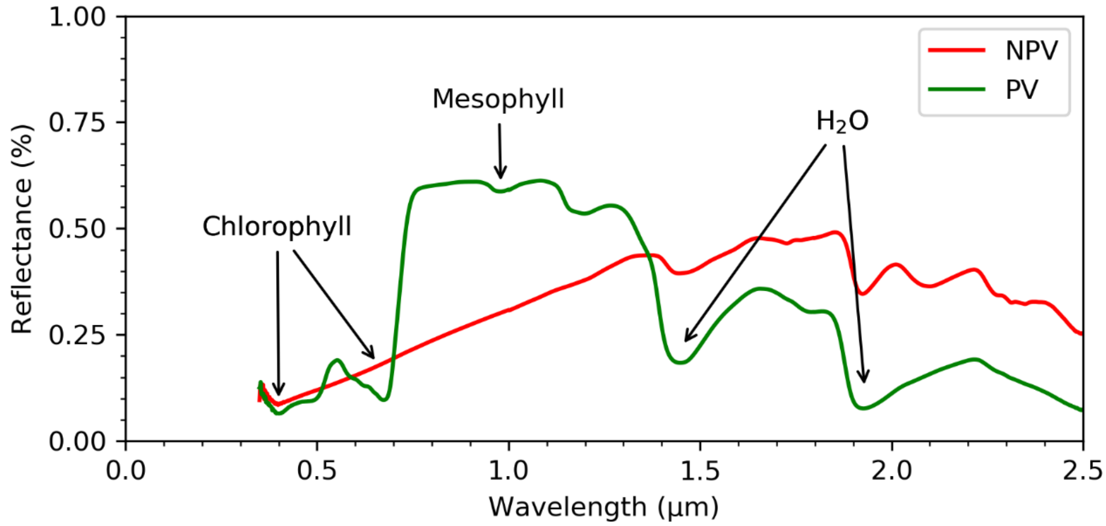


Figure 2: Average spectral curve of photosynthetic (PV) and non-photosynthetic vegetation (NPV). Mesophyll has one absorption feature centered at 1 µm; Hyper-spectral data was collected in an African woody savanna by using an ASD Fieldspec Pro by Meyer and Okin (2015).

Remote sensing models from the United States Department of Agriculture (USDA) and the United States Geological Survey (USGS) continue to map $\lambda LE$ across the continental United States at a 4 km and 1 km spatial resolution using a surface energy balance approach, radiometric temperature, and spectral vegetation indices without considering the photosynthetic pathway of the vegetation under study (Senay et al. 2013; Anderson et al. 2007). Others have used readings from remote sensing platforms to find correlations between crop coefficients (*Kc*) and vegetation indices in riparian vegetation (*Tamarix ramosissima, Phragmites australis*), pecan orchards (*Carya illinoinensis*), and maize (*Zea*) (Nagler et al. 2013; Samani et al. 2009; Kamble, Kilic, and Hubbard 2013).

The *Kc* is a dimensionless ratio of actual $\lambda LE$ to a reference evapotranspiration (*ETo*) rate from a hypothetical grass reference crop not experiencing any water stress; *ETo* is calculated through permutations of the Penman-Montieth equation at hourly or daily increments (Zotarelli and Dukes 2010; Allen et al. 1998). It is important to note that the calculation of *ETo* is influenced only by climatic conditions of wind speed, net radiation, relative humidity and air temperature. While the *Kc* should be theoretically be limited to numerical range of 0 – 1. There have been instances where actual ET rates from crops or other plant species exceed the *Kc* value of 1, such as in corn or riparian vegetation due to the vegetation's access to other sources of water (Cleverly et al. 2015; Martínez-Cob 2008).

Through various remote sensing platforms and weather parameters, an equation can be established to correlate $\lambda LE$ with a non-linear relationship between spectral indices and *Kc*; this non-linear relationship is based on a modification of Beer-Lambert's law which calculates the absorption coefficient in canopies across different plant species (Nagler et al. 2004; Nagler et al. 2013; Campbell 1986). The principles of Beer-Lambert's law are summarized in equation 2, where $\Phi$ is outgoing radiance, $\Phi_o$ is incoming radiance, $-k$ is the absorption coefficient and z is path of length of radiation traveling through the medium. In equation 2, $z$ can be representative of the canopy of vegetation and $-k$ as an absorption coefficient.

$$\Phi = \Phi_o e^{-kz} \tag{2}$$

EC measurements observe $\lambda LE$ as the sum of transpiration from vegetation ($\lambda LEc$) and evaporation from the soils ($\lambda LEs$). Various techniques have been developed to partition this energy flux into its respective quantities when direct measurements are not available. Energy balances, transpiration domes, and unmanned aerial vehicles in grasslands, orchards, and irrigated fields of alfalfa have been used to partition this energy flux (Berni et al. 2009; Hewitt, Fernald, and Samani 2018; Timmermans et al. 2007; Kool et al. 2014). The USDA Atmohspere-Land Exchange Inverse (ALEXI) uses a Two-Source Energy Balance Model to split $\lambda LE$ into its respective quantities of $\lambda LEc$ and $\lambda LEs$ through fraction of green cover derived from spectral imagery (Anderson et al. 2007; Norman, Kustas, and Humes 1995). In this project it was intended to isolate *λLE* into purely *λLEc*.

Using the International Geosphere-Biosphere Programme (IGBP) vegetation classification system, EC data were collected from Ameriflux to analyze the response of *λLEc* and $CO_2$ flux (*FC*) to spectral indices derived from various remote sensing platforms in woody savannas, grasslands, and open shrublands across a climatic gradient of the American Southwest. The land cover units of the IGBP are defined in Table 1.

| Land Cover Units | Definition |
|---|---|
| Woody Savanna | Lands with herbaceous and other understory systems, and with forest canopy cover between 30% and 60%. The forest cover height exceeds 2 m. |
| Open Shrubland | Lands with woody vegetation less than 2 m tall and with shrub canopy cover between 10% and 60%. The shrub foliage can be either evergreen or deciduous. |
| Grassland | Lands with herbaceous types of cover. Tree and shrub cover is less than 10%. |

Table 1: Vegetation classification scheme as defined by the IGBP (Friedl et al. 2002).

In this project, relationships were explored between hyper temporal observations of Normalized Difference Vegetation Index (NDVI), *FC*, and *λLEc* during the growing season in Southern California. The assessment of the non-linear relationship based on Beer-Lambert's law between *Kc* and the Enhanced Vegetation Index (EVI) was explored across all tree land cover units (Table 1) at different temporal scales (day and hour) in California, Arizona, New Mexico, and Texas. Few studies have assessed how spectral indices derived from satellite imagery affect the calibration of *λLE* estimates based on the time of capture.

This thesis has the following objectives:

1. To determine when the vegetation at all sites is most active throughout the day by using the minimum value of *FC* (the use of the minimum values is used beacseu when *FC* negative, $CO_2$ is moving from the atmosphere to the land surface) at a diurnal scale during the growing and dormant season.
2. To isolate the peak of $\lambda LEc$ at a diurnal scale during periods of clear-skies and dry days in the growing and dormant seasons of all sites.
3. To explore links between various fluxes (i.e. $\lambda LEc$, *FC*), climatic parameters (e.g. *VPD*), and the hyper temporal readings of NDVI in the Southern California sites.
4. To assess the non-linear relationship between EVI and Crop Coefficients ($K_c$) across all sites using various time steps of the EC data.

It is hypothesized that due to the prevalence of the $C_4$ photosynthetic pathway found in various grasslands of the American Southwest, the peak of $\lambda LEc$ will not line up with the time of overpass from MODIS' *Terra* and *Aqua* satellites and therefore introduce error into regional results of $\lambda LEc$ that were derived with vegetation indices. The opposite effects are expected for sites that are dominated by $C_3$ species.

# 2. METHODS

## 2.1 Description of study sites

Data from 15 EC sites were obtained from the Ameriflux Data portal for Grassland (GRA), Open Shrubland (OSH), and Woody Savanna (WSA) across a climatic gradient of the American Southwest. These land cover definitions are based on the IGBP and can be found in Table 1. In this report we defined the American Southwest as California, Arizona, New Mexico, and Texas which provided a natural rainfall gradient ranging from east to west. Data were obtained in half hourly intervals across all sites and their yearly temporal availability can be found in Table 2. Each site had an average of 10 years of data availability.

| Site | Lat, Long (WGS 84) | Land Cover | Years of data availability | State |
|---|---|---|---|---|
| Aud | 31.5907 N, 110.5104 W | GRA | 2002 - 2011 | AZ |
| Dia | 37.6773 N, 121.5296 W | GRA | 2010 - 2012 | CA |
| Scg | 33.7365 N, 117.6946 W | GRA | 2006 - 2016 | CA |
| Seg | 34.3623 N, 106.7019 W | GRA | 2007 - 2018 | NM |
| Srg | 31.7894 N, 110.8277 W | GRA | 2008 - 2018 | AZ |
| Wkg | 31.7365 N, 109.9419 W | GRA | 2004 - 2018 | AZ |
| Scs | 33.7343 N, 117.6960 W | OSH | 2006 - 2016 | CA |
| Scw | 33.6047 N, 116.4527 W | OSH | 2006 - 2016 | CA |
| Ses | 34.3349 N, 106.7442 W | OSH | 2007 - 2018 | NM |
| Whs | 31.7438 N, 110.0522 W | OSH | 2007 - 2018 | AZ |
| Wjs | 34.4255 N, 105.8615 W | OSH | 2007 - 2018 | NM |
| Fr2 | 29.9495 N, 97.9962 W | WSA | 2005 - 2008 | TX |
| Srm | 31.8214 N, 110.8661 W | WSA | 2003 - 2018 | AZ |
| Ton | 38.4316 N, 120.9660 W | WSA | 2001 - 2018 | CA |
| Mpj | 34.4384 N, 106.2377 W | WSA | 2008 - 2018 | NM |

Table 2: Ameriflux sites used in this analysis

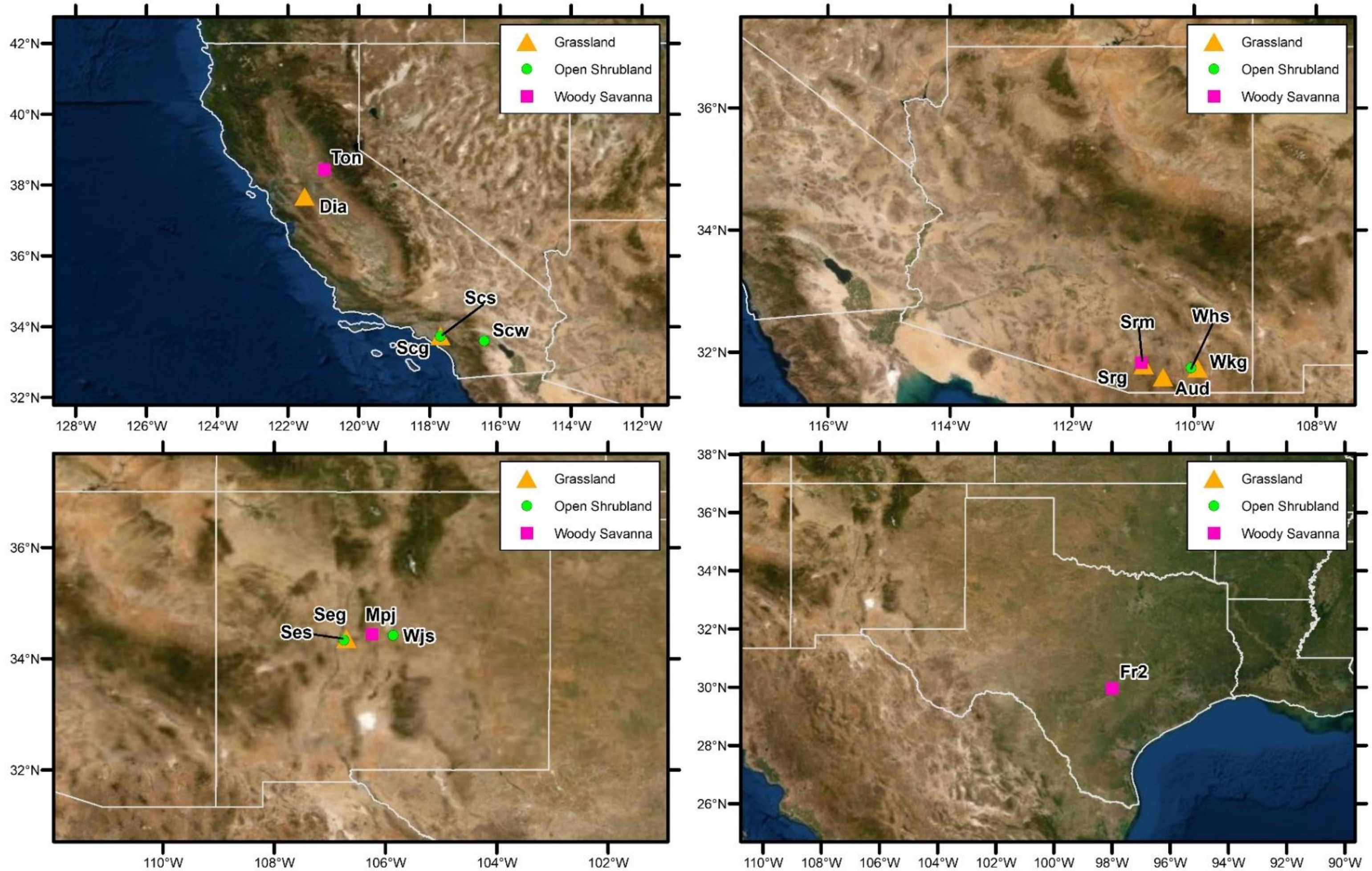


Figure 3: Map (WGS 84) of the study sites situated within the American Southwest.

### 2.1.1 CLIMATIC AND VEGETATION DESCRIPTION OF GRASSLANDS IN THE SOUTHWEST

#### 2.1.1.1 Audubon Ranch Research Site

The Audubon EC (Aud) research ranch, ~32 $km^2$ in size, was established in 1969 after grazing stopped with the goal to serve as an ecological control in studies regarding land use, endangered species, ecological restoration, and to understand the role that fire took in shaping vegetation distribution in the southwest (Bock and Bock 1986). The Aud EC site (31.5907 N, 110.5104 W; elevation 1496 m) observed an annual average of 379 mm of precipitation and an average air temperature of 15.8°C during the study period of 2004 – 2007 (Krishnan et al. 2012). Krishnan et al. (2012) described the soils at the site as sandy clay loams mixed with gravel and the vegetation being predominantly $C_4$ perennial bunch grasses (*Bouteloua gracilis, B. curtipendula,* and *Eragrostis intermedia*), non-native African lovegrasses (*E. lehmanniana* and *E. curvula*) and some scattered woody vegetation (*Mimosa aculeaticarpa*, *Astragalus nothoxys*, *Gnaphalium canescens*, *Erigeron* spp., and *Verbena gracilis*).

#### 2.1.1.2 Santa Rita Grassland Site

The Santa Rita Experimental Range was established in the early 1900s approximately 60 km south of Tucson, Arizona. In this report, there are two EC sites situated within the Santa Rita Experimental Range, Santa Rita grassland (Srg) and Santa Rita mesquite (Srm), both sides are located approximately 5 km apart. Encroachment by velvet mesquite (*Prosopis velutina*) throughout the early half of the century triggered a herbicide treatment in the 1950s to be applied to the area in order to increase herbaceous cover of *E. lehmanniana*, a non-native grass  (Whitford 1997; Mitchel P McClaran 2003). Even with the efforts to combat woody encroachment, Mesquite adapted well to the conditions of the site and across the U.S. Southwest. Mesquite is able to develop lateral root systems that can extend up to 15 m away from the canopy drip line and can enable the redistribution of deep water and shallow water throughout the year (Scott et al. 2009).

Soils at the Srg EC site (31.7894 N, 110.8277 W; elevation 1120 m) can be described as deep loamy sands, average air temperature has been recorded at 19°C with an average annual precipitation of 334 mm from 2005 – 2014, and most of the grassland is dominated by *E. lehmanniana, Digitaria californica*, *B. rothrockii*, *Muhlenbergia porter*, and several species of *Aristada* grasses (Scott et al. 2015; M. P. McClaran and Angell 2006).

#### 2.1.1.3 Walnut Gulch: Kendall Grasslands

The Walnut Gulch experimental watershed, situated near Tombstone, Arizona and ~149 $km^2$ in size, was established in 1953 the purpose of collecting long term information on the processes of runoff, hydrology, and sediment accumulation in arid lands (Renard et al. 2008). In this report, data from two EC sites situated within the watershed, Kendal Grasslands (Wkg) and Lucky Hills (Whs), approximately 10 km apart was used. $C_4$ grasses (*B. curtipendula*, *B. eriopoda*, *B. hirsute*, *Hilaria belangeri*, and *A. hamulosa*), *E. lehmanniana*, and $C_3$ shrubs (*Calliadra eriophylla*, *Dalea formosa*, *Krameria parvifolia*, *P. glandulosa*, *Yuccaelata*, and *Isocoma tenuisecta*) are the predominant vegetation of the Wkg site (31.7365 N, 109.9419 W; elevation 1531 m) (Krishnan et al. 2012). The average air temperature and average annual precipitation at the site from 2005 – 2014 is 17.3°C and 294 mm respectively while soils range from gravelly, sandy to fine sandy, and clayey loams (Scott et al. 2015; Krishnan et al. 2012)

### 2.1.1.4 Diablo grassland

The Diablo (Dia) grassland site (37.6773 N, 121.5296 W; elevation 323 m) is about 80 km southeast of San Francisco, California and falls under the jurisdiction of the of the Lawrence Livermore National Laboratory (DuBois et al. 2018). The site was established in the 1950s to aid the U.S. government's nuclear arm's race with the Soviet Union, therefore the site has not seen any grazing since 1952 (Wharton et al. 2013; Browne et al. 2017). Vegetation in the site is dominated by $C_3$ species (*Avena, Bromus* sp., *Poa secudna, Nasella* sp.) and a small amount of woody cover (*Quercus douglasii, Q. lobata*) (Preston 2006). The site is also home to the large-flowered fiddleneck (*Amsinckia grandiflora*), a rare and endangered plant species (Carlsen and Paterson 2017). The site saw an average annual precipitation of 265 mm and an average air temperature of 15.0°C (Wharton et al. 2013). Soils around the site have been described as a mixture of sand, silt, and clay (Ferry et al. 2002).

### 2.1.1.5 Southern California Grassland

The Irvine Ranch Conservancy (IRC), approximately ~375 km$^2$, was established in 2005. Grazing by cattle occurred in the site from the mid-1800s until 2002 (Kimball et al. 2014). Within the IRC, there are two EC systems, Southern California grassland (Scg) site and Southern California Coastal Sage (Scs), situated approximately 250 m apart. The Scg site (33.7365 N, 117.6946 W; elevation 470 m) is situated ~60 km southeast of Los Angeles, California. Long term climate records from 1970 – 1999 indicate an average annual precipitation of 408 mm and an average air temperature of 17.5°C (Goulden et al. 2012). Soils in the area were described as sandy loam while the dominant vegetation consist of mostly non-native annual grasses, *Bromus diandrus* and *A. fauta* (Potts et al. 2012).

#### 2.1.1.6 Sevilleta Grassland

The Sevilleta Long Term Ecological Research Network (SLTER) is situated along the central New Mexico along the Rio Grande. The site was established 1989 and is considered to be a transition zone for various ecosystems incluing riparian corridors, mountainous evergreen forests, and the Chihuahuan desert (Weiss et al. 2004). Within the SLTERN, there are two EC sites, Sevilleta Grassland (Seg) and Sevilleta Shrubland (Ses) situated 5 km apart. The Seg site (34.3623 N, 106.7019 W; elevation 1596 m) is ~80 km south of Albuquerque, New Mexico. Anderson-Teixeira et al. (2011) described the dominant vegetation of the site as *B. eriopoda*, *Gutierrezia sarothrae*, *Ceratoides lanata*, the soils as loamy sand, the mean annual precipitation at 244 mm and the mean annual air temperature as 13.4°C.

### 2.1.2 CLIMATIC AND VEGETATION DESCRIPTION OF OPEN SHRUBLANDS IN THE SOUTHWEST

#### 2.1.2.1 Southern California Coastal Sage

The Scs site (33.7343 N, 117.6960 W; elevation 475 m) is situated ~60 km southeast of Los Angeles, California within the IRC. For a description and history of the IRC, refer to section 2.1.1.5. Long term climate records from 1970 – 1999 indicate an average annual precipitation of 408 mm and an average air temperature of 17.5°C (Goulden et al. 2012). Soils in the area were described as sandy loam while the dominant vegetation consist of mostly perennial shrubs (*Artemisia californica*, *Salvia melifera*, *S. apiana*, *Malosma laurina*) (Goulden et al. 2012; Potts et al. 2012).

#### 2.1.2.2 Southern California Pinyon/Juniper Woodland

The Southern California Pinyon/Juniper Woodland (Scw) site (33.6047 N, 116.4527 W; elevation 1280 m) is situated ~170 km southeast of Los Angeles, California within the Santa Rosa and San Jacinto Mountains National Monument. Long term climate records from 1970 – 1999 indicate an average annual precipitation of 313 mm, an average air temperature of 15.8°C, and soils in the vicinity are described as coarse sandy loam (Bureau of Land Management 2003; Goulden et al. 2012). *Adenostoma fasciculatum*, *J. californica*, and *Ceanothus greggii* are the common shrub species found within the area while the presence of Pinyon (*Pinus*) is also common at this elevation (Potter 2018).

#### 2.1.2.3 Sevilleta Shrubland

The Ses site (34.3349 N, 106.7442 W; elevation 1605 m) is ~80 km south of Albuquerque, New Mexico and is part of the SLTER. For a brief description and history of the SLTER, refer to section 2.1.1.6. Anderson-Teixeira et al. (2011) described the dominant vegetation of the Ses site as creosote bush (*L. tridentata*) and *G. sarothrae*, the soils as very gravelly sandy loam, the mean annual precipitation at 244 mm and the mean annual air temperature as 13.4°C.

#### 2.1.2.4 Walnut Gulch: Lucky Hills

A brief description of the Walnut Gulch Experimental Watershed can be found in section 2.1.1.3. Woody species (*Parthenium incanum*, *Acacia constricta, L. tridentata*, and *Flourensia cernua*) are the predominant vegetation of the Wkg site (31.7438 N, 110.0522 W; elevation 1370 m) (Scott et al. 2015). Average air temperature and average annual precipitation at the site from 2008 – 2014 was reported to be 17.6°C, 285 mm respectively while soils were described as gravelly sandy loam (Scott et al. 2015).

#### 2.1.2.5 Willard Juniper Savanna

The Willard (Wjs) site, ~100 km southeast of Albuquerque, New Mexico, (34.4255 N, 105.8615 W; elevation 1926 m) was established in 2007 in order to quantify the energy cycles between the land surface and the atmosphere in arid environments (Litvak 2016a). The site is within the 7-Up 7-Down Ranch in central New Mexico and the predominant vegetation consists of *J. monosperma* and *B. gracilis* (Anderson-Teixeira et al. 2011). The site recorded a mean annual precipitation of 369 mm, an average air temperature of 11.2°C, and soils have been described as fine sandy loam (Anderson-Teixeira et al. 2011).

### 2.1.3 CLIMATIC AND VEGETATION DESCRIPTION OF WOODY SAVANNAS IN THE SOUTHWEST

#### 2.1.3.1 Mountainair Pinyon-Juniper Woodland

The Mountainair (Mpj) site, ~80 km southeast of Albuquerque, New Mexico, (34.4384 N, 106.2377 W; elevation 2196 m) was established in 2007 in order to quantify the energy cycles between the land surface and the atmosphere in arid environments (Litvak 2016b). The predominant vegetation cover at this site (> 60 %) is considered woody species (*Pinus edulis*, *J. monosperma*) with some scattered $C_4$ perennial grass (*B. gracilis*) (Litvak 2016b). The site recorded a mean annual precipitation of 420 mm, an average air temperature of 10.8°C, and soils have been described as loamy (Anderson-Teixeira et al. 2011).

#### 2.1.3.2 Freeman ranch

The site Freeman Ranch (Fr2) (29.9495 N, 97.9962 W; elevation 272 m), situated within a woody savanna ecosystem ~40 km southwest of Austin, Texas, became operational in 2004 (Litvak 2016c). The site has an approximately 870 mm of rain annually and an average air temperature of 19°C. The dominant woody vegetation in the ranch consists of mesquite juniper (*P. glandulosa*), live oak (*Q. virginiana*), and ashe juniper (*J. ashei*). Herbaceous cover by Fr2 is dominated by King Ranch bluestem (*Bothriochloa ischaemum*) and Texas wintergrass (*Stipa leucotricha*), $C_4$ and $C_3$ species respectively (Kjelgaard et al. 2008). Within the herbaceous cover of the ranch, CAM vegetation is also present in the study area in the form of prickly pear (*Opuntia engelmannii*) (Kjelgaard et al. 2008). Leaf area index measurements were approximately 0.5 from 2004 – 2006 (Kjelgaard et al. 2008). Live Oak in this ranch has heights of approximately 13 m and the basal height of approximately 4 – 5 m, while Juniper has heights of 6 - 9 m (Neuenschwander 2009). Soils in the area were described as high in clay content (Carson 2000).

#### 2.1.3.3 Santa Rita Mesquite

A brief historical description of the Santa Rita Experimental Range can be found in section 2.1.1.2. The Srm (31.8214 N, 110.8661 W; elevation 1116 m) soils can be described as deep loamy sands, an average annual precipitation of 377 mm occurred from 1937 – 2007, an average air temperature of 19.6°C, and the vegetation surrounding the site can was described as mostly as *P. veluntina*, with some scattered $C_4$ perennial grasses (*D. californica*, *M. porteri*, *B. eriopoda*, *Aristida* spp., *E. lehmanniana*) (Scott et al. 2009; Scott et al. 2015).

#### 2.1.3.4 Tonzi Ranch

The Tonzi Ranch site (Ton) (38.4316 N, 120.9660 W; elevation 177 m) is situated ~50 km Southeast of Sacramento, California. The dominant vegetation of the site (>40 %) is blue oak (*Q. douglasii*) with an average tree height of 14 m (Ma et al. 2016). Leaf Area Index estimates in the site range from 0.653 at the start of the growing season to about 1 during the later months in the year (Fisher et al. 2007). Average air temperature and annual precipitation of the site has been recorded at 15.8°C and 559 mm, respectively while soils were described as silt loams (D. Baldocchi 2016; Fisher et al. 2007).

## 2.2 Eddy Covariance

One of the key components of EC is that, unlike a traditional weather station which only measures wind-speed in two dimensions, EC is capable of measuring wind-speed in three dimensions through the use of a sonic anemometer (SA). EC also uses an infra-red gas analyzer (IRGA) to measure absorptions of known atmospheric trace gases in atmospheric eddies (e.g. $CO_2$, $H_2O$). The strength of the absorptions peaks is used to derive the concertation of a specific trace gas in an individual atmospheric eddy. Through the use of the sonic anemometer probe, IRGA, and an internal data logger that also records time, EC can measure different atmospheric eddies with different concentrations of $CO_2$, $CH_4$, $H_2O$, and temperature.

EC data can be used to derive the variables required for Equation 1, which states that $\lambda LE$ is equal to the difference of $G$ and $H$ from the total amount of $Rn$ of a surface on the earth, all measured in W/m$^2$. $Rn$ is measured through a 4-way net radiometer which measures incoming and outgoing shortwave and longwave solar radiation. The majority of shortwave radiation is energy that is incoming from the sun, while the majorty of longwave radiation is energy that the earth is emitting back into the atmosphere or space. $G$ is typically measured through a heat flux plate buried approximately 15 – 30 cm in the soil while $H$ is typically derived through the sonic temperature from a sonic anemometer and surface temperature (Burba 2013).

EC systems have the ability to record hyper-temporal data, typically at a frequency between 10 and 20 Hz. Data taken at 10 Hz produces 18,000 data points in one 30 minute window. Through Reynolds Averaging of the hyper-temporal data, equation 2 can be derived where *F* represents a concentration of an atmospheric trace gas, $\rho_a$ is the air density, *w'* is the vertical velocity, and *c'* represents the mixing ratio of the trace gas to the air density (D. D. Baldocchi 2003). *F* values that are positive indicate a movement of trace gases into the atmosphere from the land surface, negative values indicate the reverse (D. D. Baldocchi 2003).

$$F = \overline{\rho_a} * \overline{w'c'} \qquad (2)$$

Based on the vertical height from the ground of the instrumentation of an EC system and the surrounding topography, instruments with higher elevation in flat areas are able to capture a bigger footprint area of fluxes. Fetch distances for towers that range in height from 10 – 25 m and are situated within relative flat topography, is anywhere between 50 – 200 m (Cleverly et al. 2015). The specific instrumentation and set up used for these sites can be found in Table 3.

| Site | Land Cover | P | SA Model | SA Height (m) | IRGA | $H_g$ | SWC Depth (cm) | NDVI | Source Paper |
|---|---|---|---|---|---|---|---|---|---|
| Aud | GRA | Hydrol. Serv. TB3 | 81000V | 4 | ATDD Open-Path Analyzer; LI-7500 | CNR1-CM3 | 10, 20, 30 40, 60, 100 | - | Krishnan et al. 2012 |
| Dia | GRA | 260-2500-12, NovaLynx Co. | CSAT3A | 2.2 | EC150 | NR-LITE-L | - | - | Wharton et al. 2012; Maruyama and Segawa 2017 |
| Scg | GRA | - | CSAT3 | 4 | LI-7000 | CM3 | - | * | Goulden et al. 2012; Goulden et al. 2006 |
| Seg | GRA | TE525MM-L | CSAT-3 | 3.2 | LI-7500 | CNR1 | 30, 40 | - | Anderson-Teixeira et al. 2011 |
| Srg | GRA | - | - | 3.1 | LI-7500 | - | - | - | Scott et al. 2015 |
| Wkg | GRA | Belfort; TE525 | CSAT3 | 6.5 | LI-7500 | CNR1-CM3 | 5, 15 | - | Scott et al 2010; Krishnan et al 2012 |
| Scs | OSH | - | CSAT3 | 4 | LI-7000 | CM3 | - | * | Goulden et al. 2012; Goulden et al. 2006 |
| Scw | OSH | - | CSAT3 | 4 | LI-7000 | CM3 | - | * | Goulden et al. 2012; Goulden et al. 2006 |
| Ses | OSH | TE525MM-L | CSAT-3 | 3.2 | LI-7500 | CNR1 | 30, 40 | - | Anderson-Teixeira et al. 2011 |
| Whs | OSH | Belfort; TE525 | CSAT3 | 6.5 | LI-7500 | CNR2 | 5, 15, 30, 50, 75, 100 | - | Scott et al. 2010 |
| Wjs | OSH | TE525MM-L | CSAT-3 | 10.3 | LI-7500 | CNR1 | 30, 40 | - | Anderson-Teixeira et al. 2011 |
| Fr2 | WSA | Texas Electronics | CSAT-3 | 3 | LI-7500 | LI-200 | 5, 10, 20 | - | Kjelgaard et al. 2008 |
| Srm | WSA | TE525 | CSAT-3 | 7.82 | LI-7500 | CNR1 | 5, 10, 20, 30, 50, 70, 100 | - | Scott et al. 2010 |
| Ton | WSA | Texas Electronics | Model 1352 | 2 | LI-7500 | Kipp and Zonen | 5, 20, 50 | - | Xu and Baldochhi 2004 |
| Mpj | WSA | TE525MM-L | CSAT-3 | 8.2 | LI-7500 | CNR1 | 30, 40 | - | Anderson-Teixeira et al. 2011 |

Table 3: Instrumentation of EC sites used in this analysis. – Denotes that no record for instrumentation was found. * indicates that a ground NDVI sensor was used based on the design of Pontailler, Hymus, and Drake (2003).

## 2.3 Data Processing

### 2.3.1 Determining the Growing/Dormant Season

It was assumed that when *FC* was positive, the ecosystem was losing C via respiration (*RES*) and when it was negative, it was gaining carbon via photosynthesis (*GPP*). MODIS products come in temporal windows of 16 days, therefore EC data was averaged into 16-Day windows to match the remote sensing observations. When friction velocity (*U**) was below 0.1, it was assumed that turbulent fluxes did not take place and therefore these data were not incorporated into the analysis. The methodology (equation 3) outlined in (Falge et al. 2002) to calculate *S* was used, when *S* >1 it was considered the growing season and when *S* < 1, it was considered the dormant season. The data was processed with script *Seasons.py*, found in section 6.1.

$$S = \frac{GPP}{RES} \tag{3}$$

### 2.3.1 Eddy Covariance Data

The EC systems used in this report did not distinguish the quantity of $\lambda LE$ that originates in the canopy and soil. Therefore, $\lambda LE$ was isolated to represent clear-sky dry days where $\lambda LE$ would be just $\lambda LE_c$ using various climatic conditions (Williams et al. 2004) during both growing and dormant seasons. The initial data from the Ameriflux portal came in 30 minute intervals which were resampled to 1-hour mean values.

In order to obtain clear-sky conditions during dry-days, precipitation (*P*) events ($P \neq 0$ mm) were removed from the analysis. *P* data from the Southern California sites (SC) was not available. Extraterrestrial radiation ($H_o$) (equation 4) was then calculated using the FAO-56 at 1 hour intervals (Allen et al. 1998). Incoming Solar Radiation ($H_g$) was obtained from the EC systems since this would represent how much irradiance traversed through the atmosphere. At each site, $H_g$ was measured using sensors found in Table 3. Since most of the vegetation described in section 2.1 follows the $C_3$ and $C_4$ photosynthetic pathway, periods where $H_g$ was greater than or equal to 20 W/m$^2$ were filtered to represent periods with sunlight, similar to the methodologies presented in Jaksic et al. (2006) and Gallo et al. (2011). The clear-sky index ($K_t$) (equation 5) was then applied to distinguish between the different types of sky conditions (e.g. sunny, very sunny) during temperature ranges of -50°C to 50°C (Yousif, Quecedo, and Santos 2013). In this analysis, sunny and very sunny conditions were used, the thresholds can be found in table 4.

$$H_o = \frac{12(60)}{\pi} G_{sc} d_r \left[(\omega_2 - \omega_1) sin(\phi) sin(\delta) + cos(\phi) cos(\delta)\left(sin(\omega_2) - sin(\omega_1)\right)\right] \quad (4)$$

$$K_t = \frac{H_g}{H_o} \quad (5)$$

| **Sky Condition** | **Clear-sky Index ($K_t$) thresholds** |
|---|---|
| Cloudy | 0.00 – 0.20 |
| Partly Cloudy | 0.20 – 0.60 |
| Sunny | 0.60 – 0.75 |
| Very Sunny | 0.75 – 1.00 |

Table 4: Clear-sky index thresholds based on Yousif, Quecedo, and Santos (2013).

In order to nullify the effect of soil evaporation ($\lambda LE_s$), values of soil water content (SWC) were filtered between the wilting points and 33 % above the wilting point, which are based on thresholds outlined in Rawls, Brakensiek, and Saxton (1982). The parameters used for SWC can be found in table 5. SWC was not available for the sites situated within New Mexico. If a site had more than one SWC sensor, the average reading of all sensors present was used, the depths of the sensors can be found in Table 3. For the Dia and Wkg site which had a mix of two or more soil types, the average wilting and field capacity for the soils found in the sites was used. The data filtering listed in this section was performed with script *Filter_EC.py* found in section 6.2.

| USDA Soil Type | Wilting Point (-15.0 bar) | 33 % above Wilting Point |
|---|---|---|
| Sand | 0.033 | 0.052 |
| Loamy sand | 0.055 | 0.078 |
| Sandy loam | 0.095 | 0.132 |
| Loam | 0.117 | 0.268 |
| Silt loam | 0.133 | 0.199 |
| Sandy clay loam | 0.148 | 0.184 |
| Clay loam | 0.197 | 0.237 |
| Silty clay loam | 0.208 | 0.261 |
| Sandy clay | 0.239 | 0.272 |
| Silty clay | 0.250 | 0.296 |
| Clay | 0.272 | 0.313 |

Table 5: Wilting point and field capacity values of Soil Water Content ($cm^3/cm^3$) from Rawls, Brakensiek, and Saxton (1982).

### 2.3.2 Remote Sensing Data

Satellite overpass times of MODIS' *Terra* (MOD) and *Aqua* (MYD) over the EC sites was determined by using view times of the MOD11A2/MYD11A2 product (Wan, Hook, and Hulley 2015). The view times of the MOD11A2/MYD11A2 product came in local solar time ($L_S$) and were converted to Universal coordinated time (*UTC*) using equation 6 with the longitude (*Long*) of each EC site (Williamson et al. 2013; Hu et al. 2014).

$$UTC = L_s + \frac{Long}{15} \quad (6)$$

Measurements of EVI and NDVI were derived with equations 7 and 8 using the respective MODIS bands. This analysis used the MOD13Q1/MYD13Q1 product which derived an average of both EVI and NDVI using a 16-Day window (Huete et al. 1999; Didan 2015). The remote sensing data processing procedures were performed with script *Remote_Sensing_VI.py* found in section 6.4.

The SC sites also were equipped with a ground sensor that measured NDVI using photodiodes at a hyper temporal scale. The spectral range of the photodiodes for the Red and NIR on the ground sensor were 640 – 680 nm and 770 – 1000 nm with a center wavelength of 655 and 825 nm in each respective channel (Pontailler, Hymus, and Drake 2003). MODIS has a spectral range for the Red and NIR channels of 620 – 720 nm and 841 – 876 nm, respectively (Barnes and Salomonson 2017). Using this hyper-temporal data, relationships between $\lambda LEc$, $FC$, VPD and NDVI at a diurnal scale were explored. Values of NDVI from the ground sensor were compared to the two overpasses of *Terra* and *Aqua* during the growing season.

$$NDVI = \frac{(NIR - Red)}{(NIR + Red)} \tag{7}$$

$$EVI = 2.5 * \frac{(NIR - Red)}{(NIR + 6 * Red - 7.5 * Blue + 1)} \tag{8}$$

### 2.3.5 PENMAN-MONTIETH REFERENCE EVAPOTRANSPIRATION

*ETo* was calculated using weather data from the EC at hourly (mm/hr) and daily intervals (mm/day) of the resampled 16-day window (equation 11) (Zotarelli and Dukes 2010; Allen et al. 1998). In equation 11, Δ is the slope of saturation vapor pressure curve, γ is the psychometric constant, *T* is the mean air temperature (°C), $U_2$ is the average wind speed at 2 m height (m/s), *VPD* is the vapor pressure deficit, and *Cn* and *Cd* are constants for tall crop types and time steps which can be found in Table 6 (Zotarelli & Dukes, 2010).

The SC sites did not record measurements of atmospheric pressure (PA) or *VPD*. Several Arizona sites (Srg, Srm, Whs, Wkg) did not record *VPD*. The New Mexico and Ton sites did not record measurements of *G*. Equation 9 was used to calculate missing atmospheric pressure in kPa, where *z* represents the elevation above sea level of the site. Since *VPD* was calculated at different time intervals (hour and day), different approaches were used for each respective time step. The procedures to calculate *VPD* can be found in the appendix. In order to calculate *G* at the New Mexico and Ton sites, equation 1 was rearranged since *Rn*, $\lambda LE$, and *H* were measured directly (equation 10).

$$PA = 101.3\left[\frac{293 - 0.0065z}{293}\right]^{5.26} \tag{9}$$

$$G = Rn - H - LE \tag{10}$$

Wind speed measurements from all EC sites were standardized to a height of 2 m ($U_2$) using equation 12, where $U_h$ represents the measured wind speed from the EC site and *h* the height of the instrumentation (Zotarelli and Dukes 2010). It is important to note that $ET_o$ only accounts for the resistance between the vegetation canopy and atmosphere, ignoring the interaction between the soil and atmosphere. $\lambda LE_c$ was then converted to mm ($ET_a$) in order to calculate $K_c$ using equation 13. Section 2.3.5 was processed with script *Penman_Daily.py* and *Penman_Hourly.py*, found in the appendix section 6.5 and 6.6, respectively.

$$ET_o = \frac{0.408\Delta(Rn\ - G) + \gamma \frac{C_n}{T + 273} U_2(VPD)}{\Delta + \gamma(1 + C_d\, u_2)} \tag{11}$$

$$U_2 = U_h \frac{4.87}{\ln(67.8\, h - 5.42)} \tag{12}$$

$$K_c = \frac{ET_a}{ET_o} \quad (13)$$

| **Time Step** | $\boldsymbol{C_n}$ | $\boldsymbol{C_d}$ |
|---|---|---|
| Daily (mm/day) | 1600 | 0.38 |
| Hourly, daytime (mm/hr) | 66 | 0.25 |
| Hourly, nighttime (mm/hr) | 66 | 1.7 |

Table 6: Values of $C_n$ and $C_d$ for tall reference crops (Zotarelli and Dukes 2010).

### 2.3.6 Statistical Analysis and Validation

Linear and exponential regressions were performed between values of $K_c$, VIs, and *FC* using statistical modules (sciPy, Numpy) in python. Data from $K_c$, VIs, and *FC* were excluded if they were two standard-deviations away from the mean to in an effort to exclude outliers. Data plots and figures were produced using the Python matplotlib module.

# 3. RESULTS

With equation 3, the average growing ($S > 1$) and dormant ($S < 1$) seasons for each environment was calculated. The average length of the growing season in GRA was about 118 days, while OSH and WSA had an average growing season length of 176 and 231 days, respectively. Growing season length (days) and *P* during the dormant season demonstrated an exponential relationship for WSA and GRA ($R^2 = 0.43$; $p < 0.05$) (Figure 4).

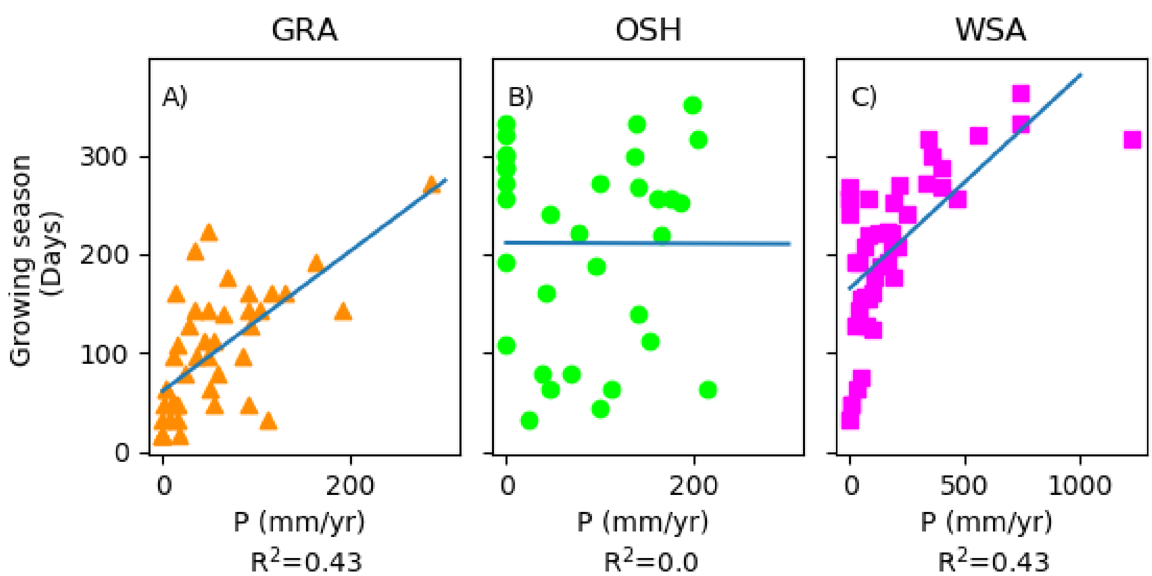


Figure 4: Linear regression between precipitation (P) and the length of the growing season across the different land cover units. Each point represents one measurement site. In GRA (A) and WSA (C) $p < 0.05$, while in OSH $p > 0.05$. GRA (A) and WSA (C) had $n = 44$, while OSH (B) had $n = 33$.

The vegetation within the GRA was active around 10:00 – 10:30 AM ($FC$ = -0.28 $gCO_2\ m^2$ hr) and 07:30 AM ($FC$ = -0.006 $gCO_2\ m^2$ hr) during the growing and dormant seasons, respectively. Values of $\lambda LEc$, VPD, and TA at the corresponding times were 80.50 ($W/m^2$), 1.14 (kPA), 19.34 (°C) and 62.88 ($W/m^2$), 0.88 (kPA), 19.34 (°C) for the growing and dormant seasons. The peak of *λLEc* was at 12:00 PM (*λLEc* = 91.27 W/$m^2$) and 07:00 AM (*λLEc* = 64.92 W/$m^2$), respectively, for the growing and dormant seasons. During the *λLEc* peaks, values of *FC*, VPD, and TA were -0.27 ($gCO_2$ $m^2$ hr), 1.31 (kPA), 20.90 (°C) and -0.002 ($gCO_2$ $m^2$ hr), 0.82 (kPA), and 14.92 (°C).

In OSH, vegetation reached its peak activity at 09:30 AM (*FC* = -0.295 $gCO_2$ $m^2$ hr) and 07:30 AM (*FC* = -0.063 $gCO_2$ $m^2$ hr) during the growing and dormant season respectively. The corresponding values of *λLEc*, VPD, and TA were 66.55 (W/$m^2$), 1.38 (kPA), 16.37 (°C) during the growing season and 86.40 (W/$m^2$), 1.24 (kPA), 16.67 (°C) during the dormant season. *λLEc* in the OSH had a maximum peak at 11:00 AM (*λLEc* = 70.25 W/$m^2$) and 12:00 PM (*λLEc* = 88.50 W/$m^2$) for the growing and dormant season respectively. *FC*, VPD, and TA at the peak of *λLEc* were -0.29 ($gCO_2$ $m^2$ hr), 1.59 (kPA), 17.90 (°C) throughout the growing season and -0.007 ($gCO_2$ $m^2$ hr), 1.72 (kPA), 19.01 (°C) during the dormant season.

WSA indicated peaks of vegetation activity at 09:30 - 10:00 AM (*FC* = -0.55 $gCO_2$ $m^2$ hr) and 03:00 PM (*FC* = -0.353 $gCO_2$ $m^2$ hr) during the growing and dormant seasons, respectively. *λLEc*, VPD, and TA during the respective peaks of vegetation activity for both growing and dormant seasons were 120.39 (W/$m^2$), 0.84 (kPA), 18.91 (°C) and 66.13 (W/$m^2$), 1.07 (kPA), 21.68 (°C), respectively. *λLEc* reached it diurnal peak at 11:00 AM (*λLEc* = 135.78 W/$m^2$) during the growing season and dormant season (*λLEc* = 158.83 W/$m^2$). During the growing season *λLEc* peak, *FC*, VPD, and TA recorded values of: -0.55 ($gCO_2$ $m^2$ hr), 0.95 (kPa), 20.15 (°C). For the dormant season *λLEc* peak, *FC*, VPD, and TA recorded values of: -0.22 ($gCO_2$ $m^2$ hr), 1.02 (kPa), 21.56 (°C). The diurnal curves of this analysis can be found in Figure 6.

Hyper-temporal readings of NDVI were available from three Southern California sites (Scs, Scg, and Scw). The relationship between NDVI and *λLEc*, *FC*, VPD resulted in low values of $R^2$ (Table 8). $R^2$ values between NDVI and *λLEc* were higher in the growing season than the dormant season with the exception of the Scw site. A linear relationshp between NDVI, *FC* ($R^2 = 0.36$; $p < 0.05$), and *λLEc* ($R^2 = 0.24$; $p < 0.05$) was found in the growing season of Scg, which had a respective classification of GRA. For more detailed results of the regression analysis and RMSE, refer to Table 8.

The average overpass time of MODIS' *Terra* (MOD) and *Aqua* (MYD) over the Southern California sites was approximately 11:00 AM and 01:00 PM, respectively. Using the average of values of satellite overpass, it was determined that the average NDVI from both satellites was 0.43, 0.46, and 0.23, while the on-ground NDVI sensors had average mean values of 0.66, 0.48, and 0.28 for each of the Southern California sites, respectively (Figure 5). Average NDVI readings from both satellites at all three sites under estimated by (~ 0.10) compared to the ground sensors. The biggest difference was observed in the Scg (GRA) site were values were satellite based NDVI values were under estimated by (~ 0.23). Peak NDVI values were registered at 04:00 PM at the Scg site, and between and 05:00 – 06:00 AM at the Scs and Scw sites based on measurements from the NDVI ground sensors.

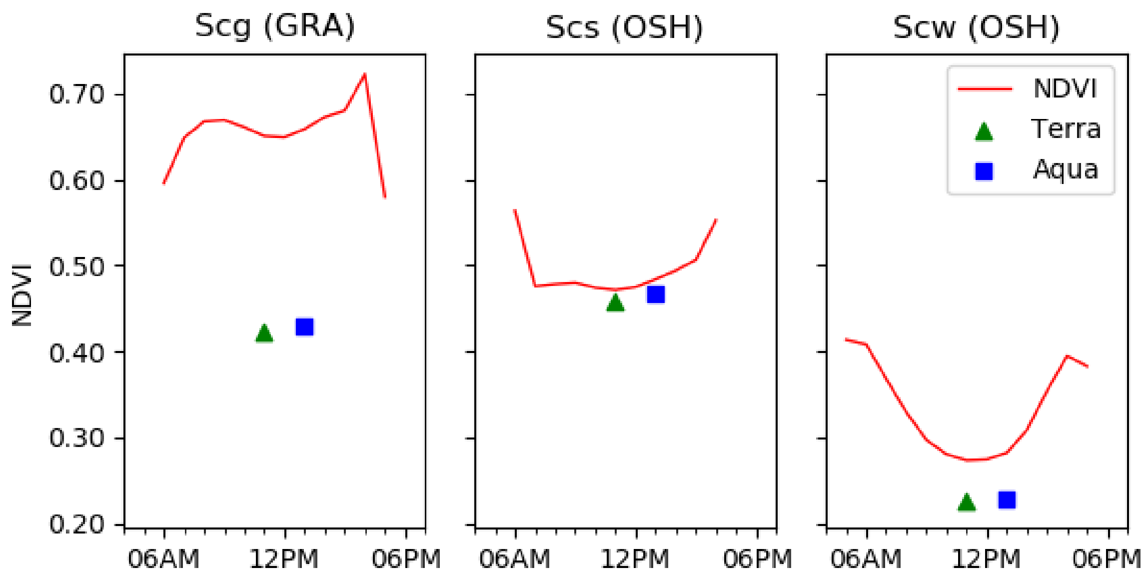


Figure 5: Diurnal profile of NDVI based on the three sites situated within Southern California. Both MODIS' satellites (*Terra* and *Aqua*) underrepresented NDVI by approximately (~ 0.10).

The linear relationship between EVI and $K_c$ during the growing season showed a strong relationship across all three land cover units when $K_c$ was calculated using inputs of mm/day during the growing season (Figure 7). Using the inputs of mm/day also resulted in statistically significant measurements across all sites ($p < 0.05$). The linear relationships were stronger when *Aqua* was used to calibrate the model with daily time steps. *Aqua* has the later overpass time situated around 01:00 – 02:00 PM across all sites. When $K_c$ values that were representative of the satellite overpass (mm/hr) were used, the relationship between EVI and $K_c$ decreased significantly compared to the daily time step values, with the exception of GRA where an increase was observed in $R^2$ was observed ($R^2 = 0.50$; $p < 0.05$) (Figure 8). The $p$ value also increased ($p > 0.05$) during the dormant season. During the dormant season, the relationship between EVI and $K_c$ seems to be relatively low or in some cases non-existent.

Using the daily time step data during the growing season, the model was calibrated to create a nonlinear relationship modeled after a modification of Beer-Lambert's law (equation 2). For the calibration of equation 14, MODIS' *Aqua* data was used since it had the highest $R^2$ based on the linear regression analysis (Figure 7). The modification of equation 2 can be seen in equation 14. The average RMSE across all sites was ± 0.0625, the lowest value of RMSE was situated in the GRA. However, the highest $R^2$ using equation 14 came from WSA ($R^2 = 0.39$), with an RMSE of ± 0.0838.

$$K_c = a\,(1 - e^{-bEVI}) - c \tag{14}$$

| **Land Cover Unit** | **Equation** | **$R^2$** | **RMSE** |
|---|---|---|---|
| GRA | $K_c = 1180.55\,(1 - e^{-0.00055*EVI}) - 0.03435$ | 0.09 | ± 0.0570 |
| OSH | $K_c = 0.60\,(1 - e^{-17.74035*EVI}) - 0.46234$ | 1.46 | ± 0.0467 |
| WSA | $K_c = 3038.71\,(1 - e^{-0.00057*EVI}) - 0.1969$ | 0.39 | ± 0.0838 |

Table 7: Results of the nonlinear relationship between EVI and $K_c$ based on the *Aqua* measurements of EVI. *Aqua* measurements are based on the daily overpass that occurs around 01:00 – 02:00 PM in all the sites.

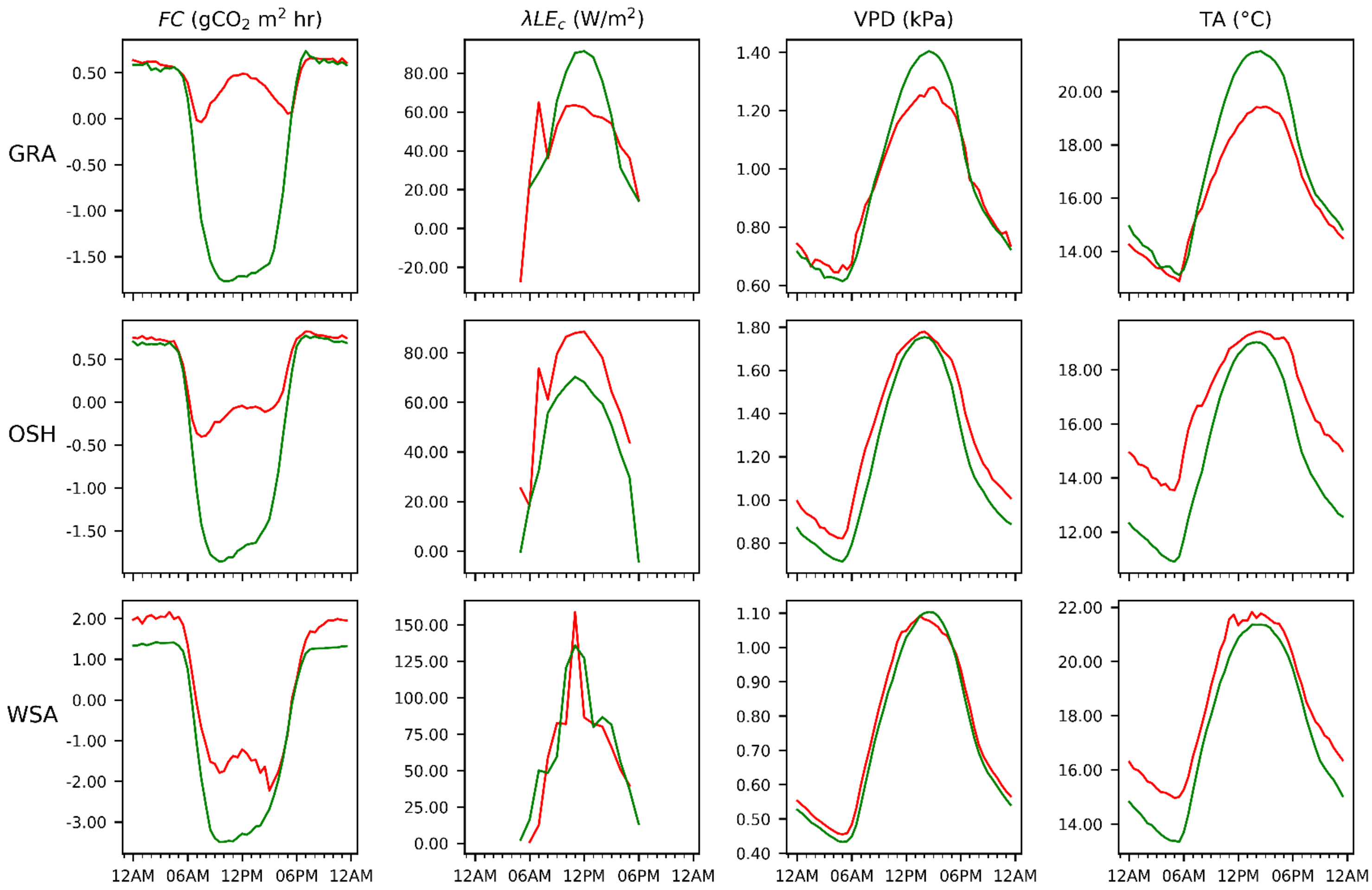


Figure 6: Diurnal curves of three land cover types (GRA, OSH, and WSA) for each respective parameter. The green curve indicates the growing season while the red curve indicates the dormant season.

| Site | Season | Variable | $R^2$ | Slope (m) | Intercept (b) | *P*-Value | RMSE |
|---|---|---|---|---|---|---|---|
| Scg (GRA) | Growing | *FC* | 0.34 | -20.83 | 8.98 | $p < 0.05$ | ± 107.9216 |
| | | $\lambda LE_c$ | 0.24 | 279.65 | -84.99 | $p < 0.05$ | ± 70.9527 |
| | | *VPD* | 0.10 | 1.57 | 0.40 | $p < 0.05$ | ± 0.6917 |
| | Dormant | *FC* | 0.03 | -10.65 | 3.92 | $p < 0.05$ | ± 41.0157 |
| | | $\lambda LE_c$ | 0.00 | -37.82 | 53.68 | $p < 0.05$ | ± 31.9389 |
| | | *VPD* | 0.04 | -3.15 | 2.38 | $p > 0.05$ | ± 0.5619 |
| Scs (OSH) | Growing | *FC* | 0.18 | -12.24 | 4.04 | $p < 0.05$ | ± 49.5803 |
| | | $\lambda LE_c$ | 0.08 | 97.95 | 0.25 | $p < 0.05$ | ± 26.579 |
| | | *VPD* | 0.04 | -1.35 | 1.94 | $p < 0.05$ | ± 0.5602 |
| | Dormant | *FC* | 0.01 | -3.04 | 1.09 | $p < 0.05$ | ± 38.3594 |
| | | $\lambda LE_c$ | 0.03 | 99.31 | -2.00 | $p < 0.05$ | ± 39.0759 |
| | | *VPD* | 0.00 | -0.87 | 2.21 | $p < 0.05$ | ± 0.8334 |
| Scw (OSH) | Growing | *FC* | 0.00 | 0.91 | -1.93 | $p > 0.05$ | ± 44.2409 |
| | | $\lambda LE_c$ | 0.01 | -64.91 | 64.2 | $p < 0.05$ | ± 35.6400 |
| | | *VPD* | 0.14 | -3.84 | 3.02 | $p < 0.05$ | ± 0.6047 |
| | Dormant | *FC* | 0.07 | -8.05 | 1.42 | $p < 0.05$ | ± 46.3111 |
| | | $\lambda LE_c$ | 0.00 | 5.43 | 44.33 | $p < 0.05$ | ± 38.2736 |
| | | *VPD* | 0.16 | -5.14 | 3.49 | $p > 0.05$ | ± 0.6698 |

Table 8: Linear regression chart for the Southern California Sites. The independent variable used in this regression is NDVI.

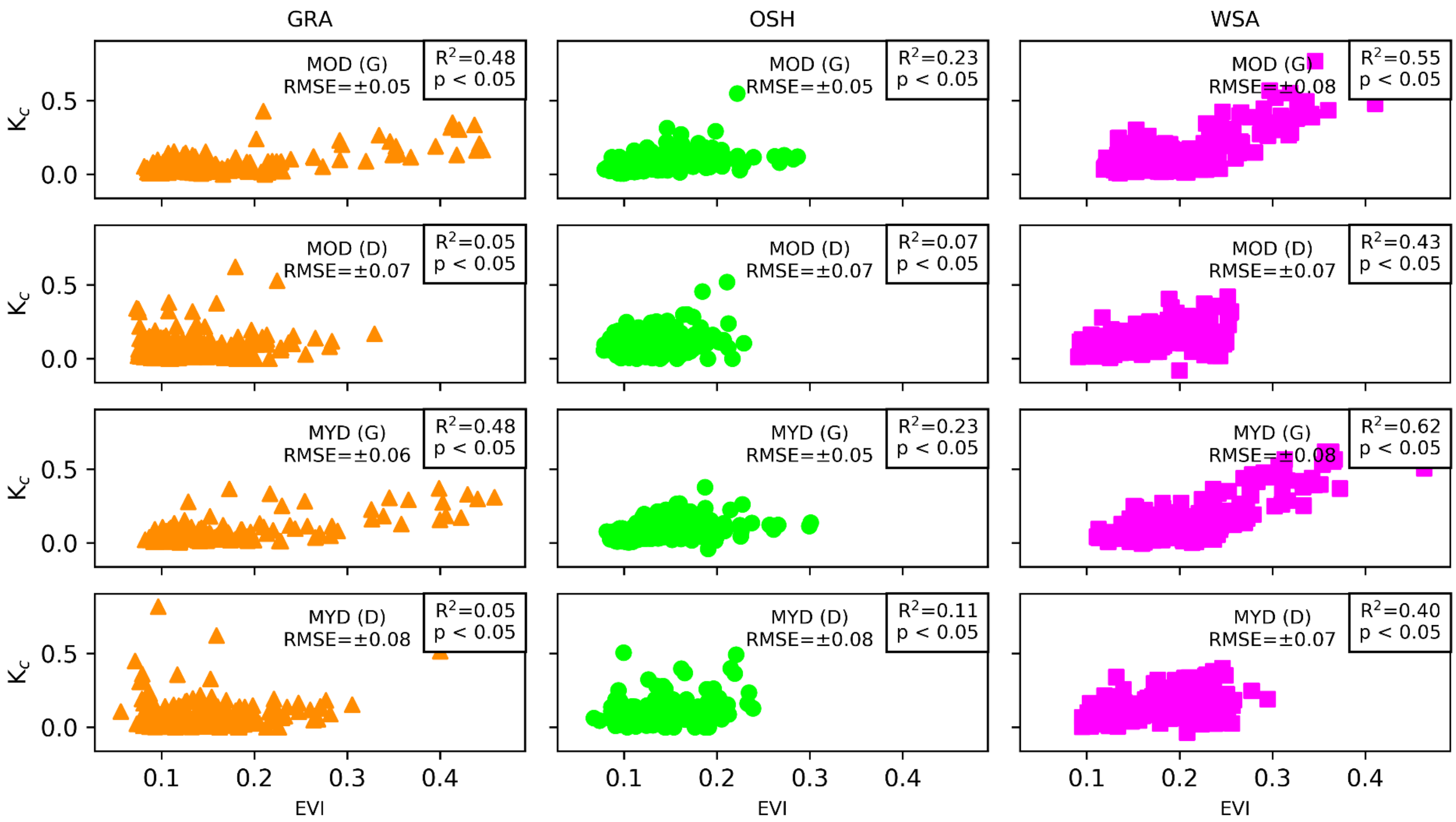


Figure 7: Results of linear regression using the daily time step data (mm/day) that was averaged in a 16-Day window. MOD corresponds to the satellite *Terra* and MYD to *Aqua*. G and D denote growing season and dormant season.

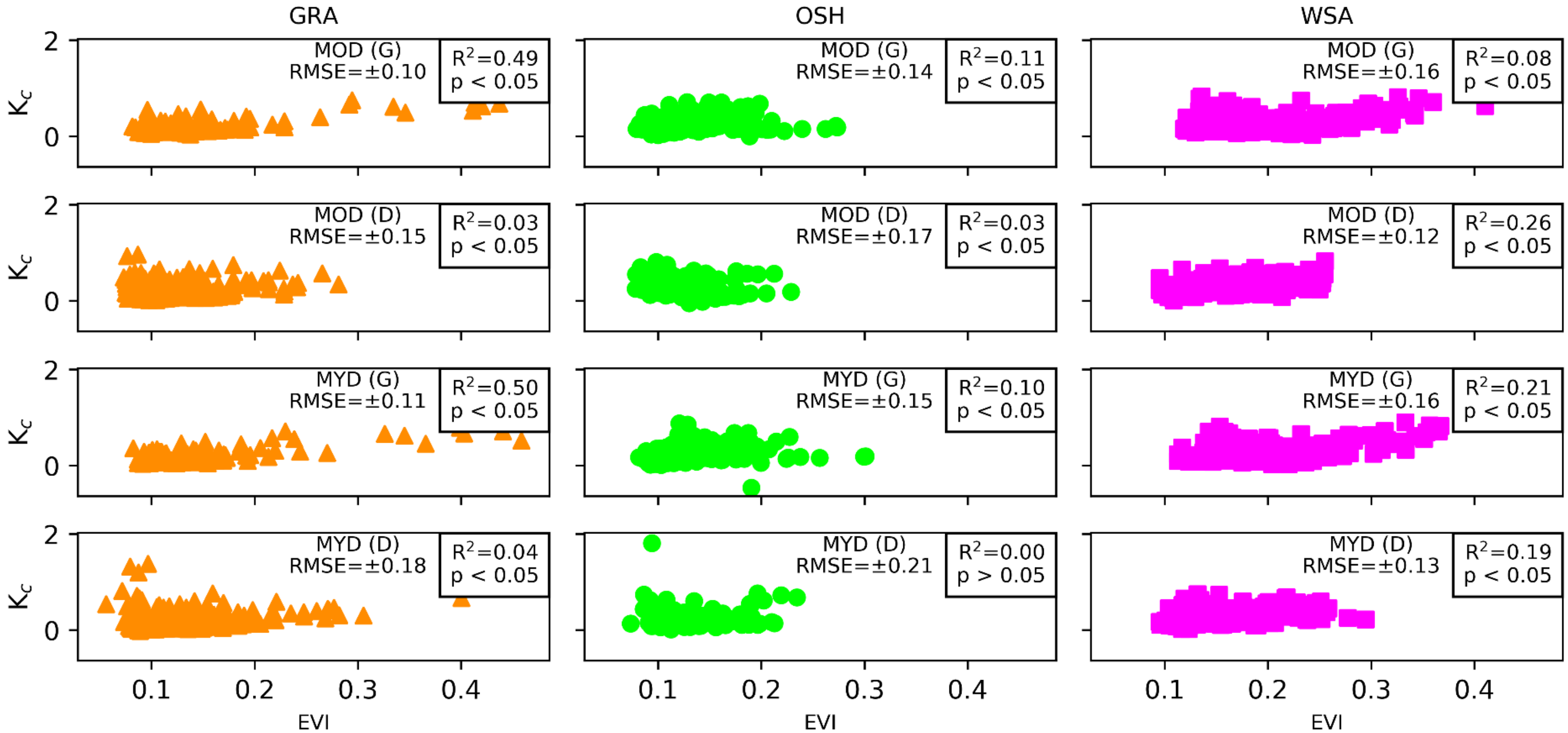


Figure 8: Results of linear regression using the corresponding time step data (mm/hr) that was recorded at the time of satellite overpass. The data averaged in a 16-Day window. MOD corresponds to the satellite *Terra* and MYD to *Aqua*. G and D denote growing season and dormant season. *Terra* and *Aqua* have an average overpass of around 1100 and 1300 across all sites, respectively.

# 4. DISCUSSION

Based on the analysis that was conducted, $P$ seems to have little impact on the length of the growing season in OSH while a logarithmic relationship seems to exits, in GRA and WSA (Figure 4). In OSH, when $P = 0$ mm/yr, the length of the growing season was still high. A possible explanation of this is missing records in $P$ or the possibility that shrubs are able to obtain water from the shallow aquifer during times of drought. Species found in OSH, such as *L. tridentata* have shown root depth between 0.15 – 0.13 m and a lateral root system that extends about 0.5 – 1.0 m from the main stem (Brisson and Reynolds 1994). The shallow root systems of *A. Californica*, found within the OSH, have allowed it to collect water during fog events from the top most soil layer, when precipitation values recorded might be null (Emery and Lesage 2015). Further analysis is required to determine the driving factors of the length of the growing season in OSH.

The non-linear relationship that was explored between EVI and $K_c$ led us to analyze this relationship through a linear regression (Figure 7 & 8). The linear relationship showed poor results ($R^2$ = 0.09, 1.46, 0.39) and therefore is not adequate in the land cover types being analyzed (Table 7). Results from the linear regression showed a strong correlation between EVI and $K_c$ during the growing season as long as the inputs were averaged into mm/day (Table 9). When the inputs were in mm/hr, which corresponded to the satellite overpass conditions, drops in $R^2$ and increase in the RMSE were observed (Figure 8).

With the use of the EC data to calculate the $K_c$, better results were observed when they were averaged over a daily period (mm/day). This type of behavior has been observed when using data from EC, since the average period reduces the variance of the data taken at ~10 Hz (Burba 2013). Results also indicated that using inputs from the earlier or later part of the day are insignificant, since the $K_c$ values are calculated in mm/day. The linear relationship that exits between EVI derived from *Aqua* and $K_c$ derived using inputs of mm/day can be modeled with the coefficients listed in Table 9.

| **Land Cover Unit** | **Equation** | $\mathbf{R^2}$ | **RMSE** |
|---|---|---|---|
| GRA | $K_c = 1.5455 * EVI + 0.0020$ | 0.55 | ± 0.05 |
| OSH | $K_c = 1.1666 * EVI + 0.0700$ | 0.46 | ± 0.06 |
| WSA | $K_c = 2.0021 * EVI - 0.0612$ | 0.67 | ± 0.11 |

Table 9: Results of the linear relationship between EVI and $K_c$ based on the *Aqua* measurements of EVI. *Aqua* measurements are based on the daily overpass that occurs around 01:00 – 02:00 PM in all the sites.

The diurnal peaks of vegetation activity for GRA, OSH, and WSA do not line up with the overpass of both MODIS' *Terra* and *Aqua*. The peak of *λLEc* for all three sites falls in between the overpasses of *Terra* and *Aqua*, with the expection of the dormant season of GRA, where the peak of *λLEc* at around 07:00 AM. The results from this analysis does not validate the hypothesis tested in this report. *λLEc* in this paper was estimated from multiple climatic conditions on the ground, therefore errors in these measurements are expected. In order to obtain a more suitable representation of the peak of *λLEc* the use of thermistors is needed, as the thermistors collect data continuously into the night and are not limited by the friction velocity of the wind (Fisher et al. 2007).

Peak NDVI values were expected to be close to the peaks of *FC* and *λLEc*, but results showed otherwise. Peak values of NDVI were observed in the early morning (05:00 – 06:00 AM) and afternoon (4:00 PM), however these could be a result of shadow effects introduced by the Earth's rotation around its axis. Shadows have an effect that alters the surface reflectance of a leaf canopy, which in turn alters the measured vegetation index. It was demonstrated that shaded leafs have higher values of NDVI due to the drop in reflectance values (Wu, Zhang, and Huang 2015). Therefore, further analysis is required to explore the links between NDVI and the various climatic parameters.

# 5. CONCLUSION

In the environments analyzed in this paper, our theory based on the modification of beer-lambert's law produced regression values that were lower than the linear analysis. A linear relationship (Table 8) was found between EVI and $K_c$ that could be used to model transpiration across a landscape scale where data is not readily available from EC. This finding is consistent to what Kamble et al. (2013) found in agricultural fields. The proposed method seems to be adequate during the growing seasons of the analyzed environments. Growing seasons can be determined using remote sensing imagery in places where EC data is not available. With the use of data from a traditional weather station that measures, relative humidity, incoming solar radiation, temperature, wind speed and with remote sensing imagery, estimates of transpiration can be derived. Limitations to this approach are that it is modeled on the Penman-Montieth equation (equation 11), which is the based on the interactions of resistances between the plant canopy and atmosphere, ignoring evaporation from the soil.

Using data from thermistors would enable the comparison of $\lambda LEc$ data to the vegetation indices directly, which can be used to partition $\lambda LE$ directly. A hyper temporal multi-spectral ground camera with the same measuring characteristics of MODIS or Landsat can be used to increase the temporal resolution and calibration efforts of the equations listed in Table 8. With the thermistors and multi spectral ground camera, the analysis listed in objective 3 can be explored directly. Another method to partition the estimates of $\lambda LEc$ is through Spectral Mixture Analysis (SMA) techniques. SMA techniques have been used in the partitioning of $\lambda LE$, which requires the use of a spectral library of the analyzed vegetation (Norman, Kustas, and Humes 1995; Anderson et al. 2007).

The linear equation presented in this analysis should be validated using independent data outside the calibration before its deployment in the calculation of water budgets. If method produces adequate results during its validation phase, it has the potential to be an application to improve water budget estimates in arid regions across the world.

# 6. APPENDIX

## 6.1 SEASONS.PY

```python
import datetime
import numpy as np
import pandas as pd
import statistics

# Path to data folder
path = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Ameriflux\\'

# US Southwest Ameriflux Sites
envs = ['GRA', 'OSH', 'WSA']
amf_dict = {'GRA': ['Aud', 'Dia', 'Scg', 'Seg', 'SRG', 'Wkg'],
            'OSH': ['Scs', 'Scw', 'Ses', 'Whs', 'Wjs'],
            'WSA': ['Fr2', 'Srm', 'Ton', 'Mpj']}

# 16 Day windows based on MODIS products
window_mod = ['001', '017', '033', '049', '065', '081', '097', '113', '129',
              '145', '161', '177', '193', '209', '225', '241', '257', '273',
              '289', '305', '321', '337', '353']

window_myd = ['009', '025', '041', '057', '073', '089', '105', '121', '137',
              '153', '169', '185', '201', '217', '233', '249', '265', '281',
              '297', '313', '329', '345', '361']

# Loop through all environments - for growing season! This is the first part of this script.
# This does not use all EC data. Only uses Friction Velocity and Carbon Flux.
fields = ['FC', 'TIMESTAMP_START', 'USTAR']

# Directory of output tables
out_table = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Seasons\Yearly\\'

# MDOIS Satellites
sats = ['MOD', 'MYD']

for env in envs:
    env_g = []
    env_d = []
    for sat in sats:
        if sat == 'MOD':
            window_16 = window_mod
            last_win = '353'
        else:
            window_16 = window_myd
            last_win = '361'
        sites = amf_dict[env]

        # Loop through the sites in the current env
        for i in sites:
            # Load the original AMF data
```

```
            table = path + env + r'\\AMF_US-' + i + '.xlsx'
            df = pd.read_excel(table, usecols=fields, skiprows=2)
            df = df.replace('', np.nan)
            df = df.replace(-9999, np.nan)
            df = df.set_index(pd.to_datetime(df['TIMESTAMP_START'], format="%Y%m%d%H%M"))

            # Keep data above Friction Velocity of 0.1 m/s
            df = df[df['USTAR'] > 0.1]

            # Keep only data of the CO2 Flux that is between 2 STDS < mean
            df = df[((df.FC - df.FC.mean()) / df.FC.std()).abs() < 2]

            # Convert from micro moles to grams/m^2/day
            # using the molecular weight of CO2; this calculates GPP
            df['gpp'] = df.FC * 0.0000010 * 44.01 * 3600 * 24
            df.gpp.loc[(df['gpp'] > 0)] = 0
            df['gpp'] = df['gpp'].abs()

            # Convert from micro moles to grams/m^2/day
            # using the molecular weight of CO2; this calculates RES
            df['res'] = df.FC * 0.0000010 * 44.01 * 3600 * 24
            df.res.loc[(df['res'] < 0)] = 0

            # Resample to daily time step
            df = df.resample('D').sum()

            # Add a column that adds its representative
            # Julian Calendar Day of Year (DOY)
            df['Yj'] = df.index
            df['Yj'] = df['Yj'].dt.strftime('%Y%j')

            # Get min and max Year of data
            df['Y'] = df.index
            df['Y'] = df['Y'].dt.strftime('%Y')

            min_yr = df.Y.min()
            max_yr = df.Y.max()

            # Resample Daily to Julian Calendar DOY to calculate
            # ratios of GPP/RES.
            # Methodology in Falge et al 2002
            df = df.set_index(pd.to_datetime(df['Yj'], format="%Y%j"))
            df.reset_index(drop=True, inplace=True)
            df['Yj'] = pd.to_datetime(df['Yj'], format='%Y%j')

            # Resample data into 16 day windows of MODIS products
            dfs = []
            for year in range(int(min_yr), int(max_yr) + 1):
                for win in window_16:

                    s = pd.Timestamp(datetime.datetime.strptime(str(year) + win + '0000', '%Y%j%H%M'))

                            # MOD conditions
                            if (year == 2000 or year == 2004 or year == 2008 or year == 2012 or year == 2016) and (
```

```
                                win == '353'):
                            jday = str(int(win) + 13)
                            e = pd.Timestamp(datetime.datetime.strptime(str(year) + jday.zfill(3) + '1159', '%Y%j%H%M'))
                        elif win == '353':
                            jday = str(int(win) + 12)
                            e = pd.Timestamp(datetime.datetime.strptime(str(year) + jday.zfill(3) + '1159', '%Y%j%H%M'))

                        # MYD codnitions
                        elif (year == 2000 or year == 2004 or year == 2008 or year == 2012 or year == 2016) and (
                                win == '361'):
                            jday = str(int(win) + 5)
                            e = pd.Timestamp(datetime.datetime.strptime(str(year) + jday.zfill(3) + '1159', '%Y%j%H%M'))
                        elif win == '361':
                            jday = str(int(win) + 4)
                            e = pd.Timestamp(datetime.datetime.strptime(str(year) + jday.zfill(3) + '1159', '%Y%j%H%M'))

                        # 16 day window
                        else:
                            jday = str(int(win) + 15)
                            e = pd.Timestamp(datetime.datetime.strptime(str(year) + jday.zfill(3) + '1159', '%Y%j%H%M'))

                        dates_mask = (df['Yj'] >= s) & (df['Yj'] <= e)
                        df2 = df.loc[dates_mask]

                        # Check whether the current MODIS 16
                        # Day window has data to compare to the EC
                        if df2.empty:
                            pass
                        else:
                            df2['period'] = df2.Yj
                            df2['period'] = df2['period'].dt.strftime('%Y')
                            df2['period'] = df2.period + win

                        dfs.append(df2)

                # Check is list are empty: if they are pass
                # other wise resample to 16 Days and save to table
                if not dfs:
                    pass
                else:
                    df3 = pd.concat(dfs, sort=True)
                    df4 = df3.groupby('period').sum()
                    df4['s'] = df4.gpp/df4.res
                    df4['gs'] = np.where(df4['s'] >= 1, 'grow', 'dorm')
                    df4['doy'] = df4.index.str.strip().str[-3:]
                    df4['Year'] = df4.index.str.strip().str[:4]

                    # Condition for leap years
                    cond_leap = (((df4['Year'] == '2000') & (df4['doy'] == last_win)) | ((df4['Year'] == '2004') & (df4['doy'] == last_win)) |
```

```
                                ((df4['Year'] == '2008') & (df4['doy'] == last_win)) | ((df4['Year'] == '2012') & (df4['doy'] == last_win)) |
                                ((df4['Year'] == '2016') & (df4['doy'] == last_win)))
                    # Condition for non leap years
                    cond = (((df4['Year'] != '2000') & (df4['doy'] == last_win)) | ((df4['Year'] != '2004') & (df4['doy'] == last_win)) |
                             ((df4['Year'] != '2008') & (df4['doy'] == last_win)) | ((df4['Year'] != '2012') & (df4['doy'] == last_win)) |
                             ((df4['Year'] != '2016') & (df4['doy'] == last_win)))
                    # Change the length of seasons using the leap conditions
                    if sat == 'MOD':
                        df4['length'] = np.where(cond_leap, 13, np.where(cond, 12, 16))
                    else:
                        df4['length'] = np.where(cond_leap, 5, np.where(cond, 4, 16))
                    # Save table to CSV file
                    df4.to_csv(out_table + r'\\US_' + i + '_' + sat + ".csv")
                # print Length of growing season days
                df4['gl'] = df4.Year + df4.gs
                df5 = df4.groupby('gl').sum()
                df5['mean_grow'] = df5.index.str.strip().str[-4:]
                df5 = df5.groupby('mean_grow').mean()
                dorm_days = df5.iloc[0]['length']
                grow_days = df5.iloc[1]['length']

                env_g.append(grow_days)
                env_d.append(dorm_days)

        avg_grow = statistics.mean(env_g)
        avg_dorm = statistics.mean(env_d)
        print('The avg growing season length for ' + env + " was " + str(avg_grow) + " days.")
        print('The avg dormant season length for ' + env + " was " + str(avg_dorm) + " days.")
```

## 6.2 FILTER_EC.PY

```python
import datetime
import numpy as np
import pandas as pd

start_time = datetime.datetime.now()

# Environments analysed (Table 1)
envs = ['GRA', 'OSH', 'WSA']

# US Southwest Ameriflux Sites (Table 2)
amf_dict = {'GRA': ['Aud', 'Dia', 'SCg', 'Seg', 'SRG', 'Wkg'],
            'OSH': ['SCs', 'SCw', 'Ses', 'Whs', 'Wjs'],
            'WSA': ['Fr2', 'Srm', 'Ton', 'Mpj']}

# Coordinates of Ameriflux sites (Table 2) in WGS 84 Decimal Degrees
coords = {'Aud': (-110.5104, 31.5907),
          'Dia': (-121.5296, 37.6773),
          'SCg': (-117.6946, 33.7365),
          'Seg': (-106.7019, 34.3623),
          'SRG': (-110.8277, 31.7894),
          'Wkg': (-109.9419, 31.7365),
          'Mpj': (-106.2377, 34.4384),
          'SCs': (-117.696, 33.7343),
          'SCw': (-116.4527, 33.6047),
          'Ses': (-106.7442, 34.3349),
          'Whs': (-110.0522, 31.7438),
          'Wjs': (-105.8615, 34.4255),
          'Ton': (-120.966, 38.4316),
          'Srm': (-110.8661, 31.8214),
          'Fr2': (-97.9962, 29.9495)}

# USDA soil types of sites
soils = {'Aud': 'Sandy clay loam',
         'Dia': 'Sand silt clay',
         'SCg': 'Sandy loam',
         'Seg': 'Loamy sand',
         'SRG': 'Loamy sand',
         'Wkg': 'Sand sandy loam',
         'Mpj': 'Loam',
         'SCs': 'Sandy loam',
         'SCw': 'Sandy loam',
         'Ses': 'Sandy loam',
         'Whs': 'Sandy loam',
         'Wjs': 'Sandy loam',
         'Ton': 'Silt loam',
         'Srm': 'Loamy sand',
         'Fr2': 'Clay'}

# (Wilting Point, Field Capacity) (Table 5)
swc_soils = {'Sand': (0.033, 0.0052),
             'Loamy sand': (0.055, 0.078),
             'Sandy loam': (0.095, 0.132),
             'Loam': (0.117, 0.168),
             'Silt loam': (0.133, 0.199),
             'Sandy clay loam': (0.148, 0.184),
```

```python
             'Clay loam': (0.197, 0.237),
             'Silty clay loam': (0.208, 0.261),
             'Sandy clay': (0.239, 0.272),
             'Silty clay': (0.25, 0.296),
             'Clay': (0.272, 0.313),
             'Sand silt clay': (0.1415, 0.174),
             'Sand sandy loam': (0.064, 0.092)}

# Loop through all sites and filter the EC data
for i in envs:
    print("Loading... " + i)
    sites = amf_dict[i]
    for site in sites:
        print("\t Processing... " + site)
        # Load Excel Sheet with data from site
        table = r'F:\C4_ET\Data\Ameriflux\\' + i + r'\\' + r'\\AMF_US-' + site + '.xlsx'

        df = pd.read_excel(table, skiprows=2)
        df = df.replace('', np.nan)
        df = df.replace(-9999, np.nan)
        df['TIMESTAMP_START'] = pd.to_datetime(df['TIMESTAMP_START'], format="%Y%m%d%H%M")
        df = df.set_index('TIMESTAMP_START')
        df = df.resample("H").mean()

        df = df.reset_index()
        df['TIMESTAMP_END'] = df.TIMESTAMP_START + datetime.timedelta(hours=1)

        # Filter out events with precipitation
        # Precipitation data not available at Southern California sites
        if site == 'SCg' or site == 'SCw' or site == 'SCs':
            pass
        else:
            df = df[df['P'] == 0]

        # Calculate Extraterrestrial Radiation (Ho)
        df['j'] = df.TIMESTAMP_START
        df['j'] = df['j'].dt.strftime('%j')
        df['doy'] = df['j'].astype(int)
        df['j'] = pd.to_datetime(df['j'], format='%j')

        # Start time obs in decimal hours
        df['h_s'] = df.TIMESTAMP_START
        df['dh_s'] = df['h_s'].dt.strftime('%H').astype(int)

        # End time obs in decimal hours
        df['h_e'] = df.TIMESTAMP_END
        df['dh_e'] = df['h_e'].dt.strftime('%H').astype(int)

        # Day angle radians
        df['Γ'] = (2 * np.pi) * ((df.doy - 1) / 365)

        # Latitude in radians
        lat_rad = coords[site][1] * (np.pi / 180)


```

```
                # Inverse relative distance Earth-Sun
                df['dr'] = 1 + (0.033 * np.cos((2 * np.pi / 365) * df.doy))

                # Solar declanation in radians
                df['δ'] = 0.409 * np.sin(((2 * np.pi) / 365) * df.doy - 1.39)

                # Mid point of observation period
                df['t'] = (df.dh_e + df.dh_s) / 2
                df['b'] = (2 * np.pi * (df.doy - 81)) / 364
                df['Sc'] = (0.1645 * np.sin(2 * df.b)) - (0.1255 * np.cos(df.b)) - (0.025 * np.sin(df.b))
                df['w'] = np.pi / 12 * ((df.t + 0.06667 * (120 - abs(coords[site][0])) + df.Sc) - 12)
                df['w1'] = df.w - (np.pi * 1) / 24
                df['w2'] = df.w + (np.pi * 1) / 24

                # Extraterrestrial radiation (Ho) in MJ m^2 hr
                df['a'] = ((df.w2 - df.w1) * np.sin(lat_rad) * np.sin(df.δ))
                df['b'] = np.cos(lat_rad) * np.cos(df.δ) * (np.sin(df.w2) - np.sin(df.w1))
                df['Ho'] = (12 * 60) / np.pi * 0.0820 * df.dr * (df.a + df.b)

                # Clear Sky Radiation
                df['Ho'] = (df.Ho / 3600) * 1000000
                df['Kt'] = df.SW_IN / df.Ho

                # Keep only very sunny conditions between -50°C to 50°C
                df = df[(df['Kt'] > 0.60) & (df['Kt'] < 1)]
                df = df[(df['TA'] >= -50) & (df['TA'] <= 50)]

                # Keep Incoming Shortwave (SWin) >= 20 W/m^2
                df = df[df['SW_IN'] >= 20]

                # Keep data above Friction Velocity of 0.1 m/s
                df = df[df['USTAR'] > 0.1]

                # Keep data between wilting point and field capacity of soils
                # Soil Water Content was not available in NM sites
                if site == 'Seg' or site == 'Ses' or site == 'Mpj' or site == 'Wjs':
                    pass
                else:
                    soil_type = soils[site]
                    wil_pnt = swc_soils[soil_type][0]
                    fld_cap = swc_soils[soil_type][1]
                    df['SWC'] = df.SWC / 100
                    df = df[(df['SWC'] > wil_pnt) & (df['SWC'] < fld_cap)]

                # Save filtered EC Data to a CSV table
                df.to_csv(r'E:\Users\fo974\OneDrive - The University of Texas at Austin\Thesis\Data\Ameriflux\\' + i + r'\\US_1HrAvg_' + site + ".csv")
```

## 6.3 Filter_EC_sc.py

```python
import datetime
import numpy as np
import pandas as pd

# Path to data
table_path = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Ameriflux\\'

# Environments analysed (Table 1)
envs = ['GRA', 'OSH']

# US Southwest Ameriflux Sites (Table 2)
amf_dict = {'GRA': ['Scg'],
            'OSH': ['Scs', 'Scw']}

# Coordinates of Ameriflux sites (Table 2) in WGS 84 Decimal Degrees
coords = {'Scg': (-117.6946, 33.7365),
          'Scs': (-117.696, 33.7343),
          'Scw': (-116.4527, 33.6047)}

# USDA soil types of sites
soils = {'Scg': 'Sandy loam',
         'Scs': 'Sandy loam',
         'Scw': 'Sandy loam'}

# (Wilting Point, Field Capacity) (Table 5)
swc_soils = {'Sandy loam': (0.095, 0.132)}

# Loop through all sites and filter the EC data
for i in envs:
    print("Loading... " + i)
    sites = amf_dict[i]
    for site in sites:
        print("\t Processing... " + site)
        # Load Excel Sheet with data from site
        table = table_path + i + r'\\' + r'\\AMF_US-' + site + '.xlsx'

        df = pd.read_excel(table, skiprows=2)
        df = df.replace('', np.nan)
        df = df.replace(-9999, np.nan)
        df['TIMESTAMP_START'] = pd.to_datetime(df['TIMESTAMP_START'], format="%Y%m%d%H%M")
        df = df.set_index('TIMESTAMP_START')
        df = df.resample("H").mean()

        df = df.reset_index()
        df['TIMESTAMP_END'] = df.TIMESTAMP_START + datetime.timedelta(hours=1)

        # Calculate Extraterrestrial Radiation (Ho)
        df['j'] = df.TIMESTAMP_START
        df['j'] = df['j'].dt.strftime('%j')
        df['doy'] = df['j'].astype(int)
        df['j'] = pd.to_datetime(df['j'], format='%j')

        # Start time obs in decimal hours
```

```python
        df['h_s'] = df.TIMESTAMP_START
        df['dh_s'] = df['h_s'].dt.strftime('%H').astype(int)

        # End time obs in decimal hours
        df['h_e'] = df.TIMESTAMP_END
        df['dh_e'] = df['h_e'].dt.strftime('%H').astype(int)

        # Day angle radians
        df['Γ'] = (2 * np.pi) * ((df.doy - 1) / 365)

        # Latitude in radians
        lat_rad = coords[site][1] * (np.pi / 180)

        # Inverse relative distance Earth-Sun
        df['dr'] = 1 + (0.033 * np.cos((2 * np.pi / 365) * df.doy))

        # Solar declanation in radians
        df['δ'] = 0.409 * np.sin(((2 * np.pi) / 365) * df.doy - 1.39)

        # Mid point of observation period
        df['t'] = (df.dh_e + df.dh_s) / 2
        df['b'] = (2 * np.pi * (df.doy - 81)) / 364
        df['Sc'] = (0.1645 * np.sin(2 * df.b)) - (0.1255 * np.cos(df.b)) - (0.025 * np.sin(df.b))
        df['w'] = np.pi / 12 * ((df.t + 0.06667 * (120 - abs(coords[site][0])) + df.Sc) - 12)
        df['w1'] = df.w - (np.pi * 1) / 24
        df['w2'] = df.w + (np.pi * 1) / 24

        # Extraterrestrial radiation (Ho) in MJ m^2 hr
        df['a'] = ((df.w2 - df.w1) * np.sin(lat_rad) * np.sin(df.δ))
        df['b'] = np.cos(lat_rad) * np.cos(df.δ) * (np.sin(df.w2) - np.sin(df.w1))
        df['Ho'] = (12 * 60) / np.pi * 0.0820 * df.dr * (df.a + df.b)

        # Clear Sky Radiation
        df['Ho'] = (df.Ho / 3600) * 1000000
        df['Kt'] = df.SW_IN / df.Ho

        # Keep only very sunny conditions between -50°C to 50°C
        df = df[(df['Kt'] > 0.60) & (df['Kt'] < 1)]
        df = df[(df['TA'] >= -50) & (df['TA'] <= 50)]

        # Keep Incoming Shortwave (SWin) >= 20 W/m^2
        df = df[df['SW_IN'] >= 20]

        # Keep data above Friction Velocity of 0.1 m/s
        df = df[df['USTAR'] > 0.1]

            # Keep data between wilting point and field capacity of soils
            # Soil Water Content was not available in NM sites
            if site == 'Seg' or site == 'Ses' or site == 'Mpj' or site == 'Wjs':
                pass
            else:
                soil_type = soils[site]
                wil_pnt = swc_soils[soil_type][0]
```

```
                    fld_cap = swc_soils[soil_type][1]
                    df['SWC'] = df.SWC / 100
                    df = df[(df['SWC'] > wil_pnt) & (df['SWC'] < fld_cap)]

                # Save filtered EC Data to a CSV table
                df.to_csv(r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Ameriflux\\' + i + r'\\US_1HrAvg_SOCA_' + site + ".csv")
```

## 6.4 REMOTE_SENSING_VI.PY

```
import os
import glob
import datetime
from osgeo import gdal
import pandas as pd

# Get driver from GDAL for GeoTiffs
driver = gdal.GetDriverByName('GTiff')

# Environments analysed (Table 1)
envs = ['GRA', 'OSH', 'WSA']

# US Southwest Ameriflux Sites (Table 2)
amf_dict = {'GRA': ['Aud', 'Dia', 'SCg', 'Seg', 'SRG', 'Wkg'],
            'OSH': ['SCs', 'SCw', 'Ses', 'Whs', 'Wjs'],
            'WSA': ['Fr2', 'Srm', 'Ton', 'Mpj']}

# Coordinates of Ameriflux sites (Table 2) in WGS 84 Decimal Degrees
coords = {'Aud': (-110.5104, 31.5907),
          'Dia': (-121.5296, 37.6773),
          'SCg': (-117.6946, 33.7365),
          'Seg': (-106.7019, 34.3623),
          'SRG': (-110.8277, 31.7894),
          'Wkg': (-109.9419, 31.7365),
          'Mpj': (-106.2377, 34.4384),
          'SCs': (-117.696, 33.7343),
          'SCw': (-116.4527, 33.6047),
          'Ses': (-106.7442, 34.3349),
          'Whs': (-110.0522, 31.7438),
          'Wjs': (-105.8615, 34.4255),
          'Ton': (-120.966, 38.4316),
          'Srm': (-110.8661, 31.8214),
          'Fr2': (-97.9962, 29.9495)}

# # List of MODIS products that we processed data
mod_prods = ["MOD11A2", "MYD11A2", "MOD13Q1", "MYD13Q1"]

# Loop to process GeoTiffs. GeoTiffs were extracted and processed with GDAL
# from HDF files. Script can be found in Appendix (MODIS_HDF_process.py)
# This creates data output tables in CSV format
for i in envs:
    print("Loading... " + i)
    sites = amf_dict[i]
    for site in sites:
        for mod_prod in mod_prods:
```

```
            if mod_prod == 'MOD11A2' or mod_prod == 'MYD11A2':
                tiffs = glob.glob(r'F:\C4_ET\Data\GIS\\' + mod_prod + r'\Tiffs\*.tif')
                out_table = r'E:\Users\fo974\OneDrive - The University of Texas at Austin\Thesis\Data\RS_Data\Time\\' + site + "_" + mod_prod + ".csv"
                col_name = 'Time'
                scale = 0.1
            else:
                tiffs = glob.glob(r'F:\C4_ET\Data\GIS\\' + mod_prod + r'\Tiffs\*gcs_ndvi_*.tif')
                out_table = r'E:\Users\fo974\OneDrive - The University of Texas at Austin\Thesis\Data\RS_Data\NDVI\\' + site + "_" + mod_prod + '_NDVI.csv'
                col_name = 'NDVI'
                scale = 0.0001
            print("\t " + mod_prod + "... " + site)
            # These lists are where the data are temp stored
            sig = []
            dates = []
            for tiff in tiffs:
                # Get dates from tif files names
                file_name = os.path.basename(tiff).split(".")[0]
                date = os.path.basename(file_name).split("_")[2]
                date = datetime.datetime.strptime(date, '%Y%j')
                dates.append(date)

                # Open TIF file and  the first band
                dataset = gdal.Open(tiff)
                band = dataset.GetRasterBand(1)

                # Determine Columns and Rows of raster
                cols = dataset.RasterXSize
                rows = dataset.RasterYSize

                # Transform the raster to the Projection
                transform = dataset.GetGeoTransform()
                xOrigin = transform[0]
                yOrigin = transform[3]
                pixelWidth = transform[1]
                pixelHeight = -transform[5]

                # Read tif file as Numpy array
                data = band.ReadAsArray(0, 0, cols, rows)
                col = int((coords[site][0] - xOrigin) / pixelWidth)
                row = int((yOrigin - coords[site][1]) / pixelHeight)

                # Append Pixel values and scale them using MODIS values
                sig.append((data[row][col]) * scale)
            # Load list into Pandas dataframe
            df = pd.DataFrame({col_name: sig})
            df.index = pd.DatetimeIndex(dates)
            # Convert Local Solar Time from MOD11A2/MYD1A2 to Local Time
            if mod_prod == 'MOD11A2' or mod_prod == 'MYD11A2':
                df['UTC'] = df.Time + (abs(coords[site][0]) / 15)

            # Save data to CSV table
```

```
            df.to_csv(out_table)
```

### 6.5 PENMAN_DAILY.PY

```
import datetime
import numpy as np
import pandas as pd

# Path to one drive data folder
path = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Ameriflux\\'
out_table = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\ETo\\'

# Environments analysed (Table 1)
envs = ['GRA', 'OSH', 'WSA']

# US Southwest Ameriflux Sites (Table 2)
amf_dict = {'GRA': ['Aud', 'Dia', 'SCg', 'Seg', 'SRG', 'Wkg'],
            'OSH': ['SCs', 'SCw', 'Ses', 'Whs', 'Wjs'],
            'WSA': ['Fr2', 'Srm', 'Ton', 'Mpj']}

# Elevations of SC sites
elev = {'SCw': 1281,
        'SCg': 465,
        'SCs': 470}

# Height (m) of instrumentation of EC Sites
ins_height = {'Aud': 4,
              'Dia': 2.2,
              'SCg': 4,
              'Seg': 3.2,
              'SRG': 3.10,
              'Wkg': 6.5,
              'Mpj': 8.2,
              'SCs': 4,
              'SCw': 4,
              'Ses': 3.2,
              'Whs': 6.5,
              'Wjs': 10.3,
              'Ton': 2,
              'Srm': 7.82,
              'Fr2': 3}

# 16 Day windows based on MODIS products
window_mod = ['001', '017', '033', '049', '065', '081', '097', '113', '129',
              '145', '161', '177', '193', '209', '225', '241', '257', '273',
              '289', '305', '321', '337', '353']

window_myd = ['009', '025', '041', '057', '073', '089', '105', '121', '137',
              '153', '169', '185', '201', '217', '233', '249', '265', '281',
              '297', '313', '329', '345', '361']

windows = [window_mod, window_myd]

```

```python
for env in envs:
    print("Analyzing Environment: " + env)
    sites = amf_dict[env]

    for window_16 in windows:
        if window_16 == window_mod:
            sat = 'mod'
        else:
            sat = 'myd'

        for site in sites:
            table = path + env + r'\\US_1HrAvg_' + site + r'.csv'

            print("\t Loading... " + site)
            # SC sites have no VPD nor PA (kPa)
            if site == 'SCg' or site == 'SCs' or site == 'SCw':
                fields = ['TA', 'WS', 'RH', 'NETRAD', 'TIMESTAMP_START', 'LE', 'H', 'G']
                df = pd.read_csv(table, usecols=fields, encoding='latin1')

                # Calculate PA (kPa) using Zotarelli et al 2010
                df['PA'] = 101.3 * ((293 - 0.0065 * elev[site]) / 293) ** 5.26

            # SRG, Whs, Srm, Wkg site has no VPD data
            elif site == 'SRG' or site == 'Wkg' or site == 'Whs' or site == 'Srm':
                fields = ['TA', 'WS', 'RH', 'NETRAD', 'PA', 'TIMESTAMP_START', 'LE', 'H', 'G']
                df = pd.read_csv(table, usecols=fields, encoding='latin1')

            # New Mexico data has no Ground Heat Flux Data
            elif site == "Mpj" or site == "Seg" or site == "Ses" or site == "Wjs" \
                    or site == "Ton":
                fields = ['TA', 'WS', 'RH', 'NETRAD', 'PA', 'TIMESTAMP_START', 'LE', 'H', 'VPD']
                df = pd.read_csv(table, usecols=fields, encoding='latin1')

            # Rest of the sites have required components measured at EC site
            else:
                fields = ['TA', 'WS', 'RH', 'NETRAD', 'PA', 'TIMESTAMP_START', 'G', 'LE', 'VPD']
                df = pd.read_csv(table, usecols=fields, encoding='latin1')

            # Replace no data values
            df = df.replace('', np.nan)
            df = df.replace(-9999, np.nan)
            df = df.set_index(pd.to_datetime(df['TIMESTAMP_START'], dayfirst=True))

            # Set values of Ground Heat flux to Joules/m^2/Hr
            # Some data does not have G (NM DATA & Tonzi Ranch)
            if site == 'Mpj' or site == 'Seg' or site == 'Ses' or site == 'Wjs' or site == 'Ton':
                df['G_J'] = (df.NETRAD - df.H - df.LE) * 3600
            else:
                df['G_J'] = df.G * 3600
```

```
# Convert Data to Joules/m^2/Hr
df['Rn_J'] = df.NETRAD * 3600
df['ET_J'] = df.LE * 3600

# Convert Joules to Mega-Joules per day
df['Rn_MJ'] = df.Rn_J / 1000000
df['G_MJ'] = df.G_J / 1000000

# Convert LE to MJ/m^2/day and to mm/day
df['ETa'] = (df.ET_J / 1000000) * 0.408

# Create and set variables equal to data in order to sort
# SC does not have VPD; therefore we will calculate it
# at a daily time step
if site == 'SCg' or site == 'SCs' or site == 'SCw' or site == 'SRG' or site == 'Wkg' or site == 'Whs' \
        or site == 'Srm':
    df['Tmean'] = df.TA
    df['Tmin'] = df.TA
    df['Tmax'] = df.TA
    df['RH_min'] = df.RH
    df['RH_max'] = df.RH
    df['U2'] = df.WS
    df['PA'] = df.PA
    df['ETa'] = df.ETa
    df['Rn_MJ'] = df.Rn_MJ
    df['G_MJ'] = df.G_MJ

    df1 = df.resample("D").agg({'Rn_J': 'sum',
                                'G_J': 'sum',
                                'Tmean': 'mean',
                                'Tmin': 'min',
                                'Tmax': 'max',
                                'RH_min': 'min',
                                'RH_max': 'max',
                                'U2': 'mean',
                                'PA': 'mean',
                                'ETa': 'sum',
                                'Rn_MJ': 'sum',
                                'G_MJ': 'sum'})
else:
    df['Tmean'] = df.TA
    df['Tmin'] = df.TA
    df['Tmax'] = df.TA
    df['RH_min'] = df.RH
    df['RH_max'] = df.RH
    df['U2'] = df.WS
    df['PA'] = df.PA
    df['VPD'] = df.VPD
    df['ETa'] = df.ETa
    df['Rn_MJ'] = df.Rn_MJ

    df1 = df.resample("D").agg({'Rn_J': 'sum',
                                'G_J': 'sum',
                                'Tmean': 'mean',
                                'Tmin': 'min',
```

```
                                                    'Tmax': 'max',
                                                    'RH_min': 'min',
                                                    'RH_max': 'max',
                                                    'U2': 'mean',
                                                    'PA': 'mean',
                                                    'VPD': 'mean',
                                                    'ETa': 'sum',
                                                    'Rn_MJ': 'sum',
                                                    'G_MJ': 'sum'})

                    # Calculate the Psychometric Constant
                    df1['γ'] = 0.000665 * df1.PA

                    # Slope of saturation vapor pressure curve
                    df1['Δ'] = (4098 * (0.6108 * np.exp((17.27 * df1.Tmean) / (df1.Tmean + 237.3)))) / ((df1.Tmean + 237.3) ** 2)

                    # Calculate the VPD (kPa) for SC sites using Zotarelli et al 2010
                    if site == 'SCg' or site == 'SCs' or site == 'SCw' or site == 'SRG' or site == 'Wkg' or site == 'Whs' \
                            or site == 'Srm':
                        df1['e_tmax'] = 0.6108 * np.exp((17.27 * df1.Tmax) / (df1.Tmax + 237.3))
                        df1['e_tmin'] = 0.6108 * np.exp((17.27 * df1.Tmin) / (df1.Tmin + 237.3))
                        df1['es'] = (df1.e_tmax + df1.e_tmin) / 2
                        df1['ea'] = ((df1.e_tmin * (df1.RH_max / 100)) + (df1.e_tmax * (df1.RH_min / 100))) / 2
                        df1['VPD'] = df1.es - df1.ea

                    else:
                        df1['VPD'] = df1.VPD * 0.1

                    # Standardize wind measurements to 2 m
                    df1['U2'] = df1.U2 * (4.87/np.log(67.8 * ins_height[site] - 5.42))

                    # Calculate ETo and Kc
                    df1['ETo'] = ((0.408 * df1.Δ * (df1.Rn_MJ - df1.G_MJ)) + (
                            df1.γ * (1600 / (df1.Tmean + 273)) * df1.U2 * df1.VPD)) / (df1.Δ + (df1.γ * (1 + (0.38 * df1.U2))))
                    df1['Kc'] = df1.ETa / df1.ETo
                    df1 = df1.dropna()

                    # Add a column with Year + DOY
                    df1['Yj'] = df1.index
                    df1['Yj'] = df1['Yj'].dt.strftime('%Y%j')
                    df1['Yj'] = pd.to_datetime(df1['Yj'], format="%Y%j")

                    # Get min and max Year of data
                    df1['Y'] = df1.index
                    df1['Y'] = df1['Y'].dt.strftime('%Y')

                    min_yr = df1.Y.min()
                    max_yr = df1.Y.max()

```

```
                    # Resample data into 16 day windows of MODIS products
                    dfs = []
                    for year in range(int(min_yr), int(max_yr) + 1):
                        for count, win in enumerate(window_16):
                            s = pd.Timestamp(datetime.datetime.strptime(str(year) + win + '0000', '%Y%j%H%M'))

                            # MOD conditions
                            if (year == 2000 or year == 2004 or year == 2008 or year == 2012 or year == 2016) and (
                                    win == '353'):
                                jday = str(int(win) + 13)
                                e = pd.Timestamp(datetime.datetime.strptime(str(year) + jday.zfill(3) + '1159', '%Y%j%H%M'))
                            elif win == '353':
                                jday = str(int(win) + 12)
                                e = pd.Timestamp(datetime.datetime.strptime(str(year) + jday.zfill(3) + '1159', '%Y%j%H%M'))

                            # MYD codnitions
                            elif (year == 2000 or year == 2004 or year == 2008 or year == 2012 or year == 2016) and (
                                    win == '361'):
                                jday = str(int(win) + 5)
                                e = pd.Timestamp(datetime.datetime.strptime(str(year) + jday.zfill(3) + '1159', '%Y%j%H%M'))
                            elif win == '361':
                                jday = str(int(win) + 4)
                                e = pd.Timestamp(datetime.datetime.strptime(str(year) + jday.zfill(3) + '1159', '%Y%j%H%M'))

                            # 16 day window
                            else:
                                jday = str(int(win) + 15)
                                e = pd.Timestamp(datetime.datetime.strptime(str(year) + jday.zfill(3) + '1159', '%Y%j%H%M'))

                            dates_mask = (df1['Yj'] >= s) & (df1['Yj'] <= e)
                            df2 = df1.loc[dates_mask]
                            df2['period'] = str(year) + win
                            dfs.append(df2)

                    # Check is list are empty: if they are pass
                    # other wise resample to 16 Days and save to table
                    if not dfs:
                        pass
                    else:
                        df3 = pd.concat(dfs)
                        df4 = df3.groupby('period').mean()
                        df4.to_csv(out_table + r'\\US_Daily_' + site + '_' + sat + ".csv")
```

## 6.6 Penman_Hourly.py

```python
import datetime
import numpy as np
import pandas as pd

# Path to one drive data folder
path = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Ameriflux\\'
out_table = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\ETo\\'

# Environments analysed (Table 1)
envs = ['GRA', 'OSH', 'WSA']

# US Southwest Ameriflux Sites (Table 2)
amf_dict = {'GRA': ['Aud', 'Dia', 'SCg', 'Seg', 'SRG', 'Wkg'],
            'OSH': ['SCs', 'SCw', 'Ses', 'Whs', 'Wjs'],
            'WSA': ['Fr2', 'Srm', 'Ton', 'Mpj']}

# Elevations of SC sites
elev = {'SCw': 1281,
        'SCg': 465,
        'SCs': 470}

# Height (m) of instrumentation of EC Sites
ins_height = {'Aud': 4,
              'Dia': 2.2,
              'SCg': 4,
              'Seg': 3.2,
              'SRG': 3.10,
              'Wkg': 6.5,
              'Mpj': 8.2,
              'SCs': 4,
              'SCw': 4,
              'Ses': 3.2,
              'Whs': 6.5,
              'Wjs': 10.3,
              'Ton': 2,
              'Srm': 7.82,
              'Fr2': 3}

# 16 Day windows based on MODIS products
window_mod = ['001', '017', '033', '049', '065', '081', '097', '113', '129',
              '145', '161', '177', '193', '209', '225', '241', '257', '273',
              '289', '305', '321', '337', '353']

window_myd = ['009', '025', '041', '057', '073', '089', '105', '121', '137',
              '153', '169', '185', '201', '217', '233', '249', '265', '281',
              '297', '313', '329', '345', '361']

windows = [window_mod, window_myd]

for env in envs:
    print("Analyzing Environment: " + env)
    sites = amf_dict[env]

```

```
    for window_16 in windows:
        if window_16 == window_mod:
            sat = 'mod'
        else:
            sat = 'myd'

        for site in sites:
            # This uses the 30 min data from ameriflux!
            table = path + env + r'\\AMF_US-' + site + r'.xlsx'

            print("\t Loading... " + site)
            # SC sites have no VPD nor PA (kPa)
            if site == 'SCg' or site == 'SCs' or site == 'SCw':
                fields = ['TA', 'WS', 'RH', 'NETRAD', 'TIMESTAMP_START', 'LE', 'H', 'G']
                df = pd.read_excel(table, usecols=fields, skiprows=2)

                # Calculate PA (kPa) using Zotarelli et al 2010
                df['PA'] = 101.3 * ((293 - 0.0065 * elev[site]) / 293) ** 5.26

            # SRG, Whs, Srm, Wkg site has no VPD data
            elif site == 'SRG' or site == 'Wkg' or site == 'Whs' or site == 'Srm':
                fields = ['TA', 'WS', 'RH', 'NETRAD', 'PA', 'TIMESTAMP_START', 'LE', 'H', 'G']
                df = pd.read_excel(table, usecols=fields, skiprows=2)

            # New Mexico data has no Ground Heat Flux Data
            elif site == "Mpj" or site == "Seg" or site == "Ses" or site == "Wjs" \
                    or site == "Ton":
                fields = ['TA', 'WS', 'RH', 'NETRAD', 'PA', 'TIMESTAMP_START', 'LE', 'H', 'VPD']
                df = pd.read_excel(table, usecols=fields, skiprows=2)

            # Rest of the sites have required components measured at EC site
            else:
                fields = ['TA', 'WS', 'RH', 'NETRAD', 'PA', 'TIMESTAMP_START', 'G', 'LE', 'VPD']
                df = pd.read_excel(table, usecols=fields, skiprows=2)

            # Replace no data values
            df = df.replace('', np.nan)
            df = df.replace(-9999, np.nan)
            df = df.dropna()
            df = df.set_index(pd.to_datetime(df['TIMESTAMP_START'], format="%Y%m%d%H%M"))

            # Set values of Ground Heat flux to Joules/m^2/Hour
            # This is required since some data does not have Ground Heat Flux Data (NM DATA & Toni Ranch)
            if site == 'Mpj' or site == 'Seg' or site == 'Ses' or site == 'Wjs' or site == 'Ton':
                df['G_J'] = (df.NETRAD - df.H - df.LE) * 3600
            else:
                        df['G_J'] = df.G * 3600

```

```
# Convert Data to Joules/m^2/Hour
df['Rn_J'] = df.NETRAD * 3600
df['ET_J'] = df.LE * 3600

# Convert Joules to MJ/hour
df['Rn_MJ'] = df.Rn_J / 1000000
df['G_MJ'] = df.G_J / 1000000

# Convert LE to MJ/hour and to mm/hour
df['ETa'] = (df.ET_J / 1000000) * 0.408

# Create and set variables equal to data in order to sort
# SC does not have VPD; therefore we will calculate it at
# a daily time step
if site == 'SCg' or site == 'SCs' or site == 'SCw' or site == 'SRG' or site == 'Wkg' or site == 'Whs' \
        or site == 'Srm':
    df['Tmean'] = df.TA
    df['RHmean'] = df.RH
    df['U2'] = df.WS
    df['PA'] = df.PA
    df['ETa'] = df.ETa
    df['Rn_MJ'] = df.Rn_MJ
    df['G_MJ'] = df.G_MJ

    df1 = df.resample("H").agg({'Tmean': 'mean',
                                'RHmean': 'mean',
                                'U2': 'mean',
                                'PA': 'mean',
                                'ETa': 'sum',
                                'Rn_MJ': 'sum',
                                'G_MJ': 'sum'})
else:
    df['Tmean'] = df.TA
    df['RHmean'] = df.RH
    df['U2'] = df.WS
    df['PA'] = df.PA
    df['VPD'] = df.VPD
    df['ETa'] = df.ETa
    df['Rn_MJ'] = df.Rn_MJ

    df1 = df.resample("H").agg({'Tmean': 'mean',
                                'RHmean': 'mean',
                                'U2': 'mean',
                                'PA': 'mean',
                                'VPD': 'mean',
                                'ETa': 'sum',
                                'Rn_MJ': 'sum',
                                'G_MJ': 'sum'})

# Standardize wind measurements to 2 m
df1['U2'] = df1.U2 * (4.87 / np.log(67.8 * ins_height[site] - 5.42))

# Calculate the Psychometric Constant
df1['γ'] = 0.000665 * df1.PA

```

```python
# Slope of saturation vapor pressure curve
df1['Δ'] = (4098 * (0.6108 * np.exp((17.27 * df1.Tmean) / (df1.Tmean + 237.3)))) / (
            (df1.Tmean + 237.3) ** 2)

# Calculate the VPD (kPa) for SC sites using
# Zotarelli et al 2010
if site == 'SCg' or site == 'SCs' or site == 'SCw' or site == 'SRG' or site == 'Wkg' or site == 'Whs' \
        or site == 'Srm':
    df1['etr'] = (0.6108 * np.exp((17.27 * df1.Tmean) / (df1.Tmean + 237.3)))
    df1['ea'] = df1.etr * (df1.RHmean / 100)
    df1['VPD'] = df1.etr - df1.ea

else:
    df1['VPD'] = df1.VPD * 0.1

# Calculate ETo and Kc
df1['ETo'] = ((0.408 * df1.Δ * (df1.Rn_MJ - df1.G_MJ)) + (
        df1.γ * (66 / (df1.Tmean + 273)) * df1.U2 * df1.VPD)) / (
                     df1.Δ + (df1.γ * (1 + (0.38 * df1.U2))))
df1['Kc'] = df1.ETa / df1.ETo
df1 = df1.dropna()

# Add a column with Year + DOY + Hour
df1['Yjh'] = df1.index
df1['Yjh'] = df1['Yjh'].dt.strftime('%Y%j%H%M')
df1['Yjh'] = pd.to_datetime(df1['Yjh'], format="%Y%j%H%M")

# Get min and max Year of data
df1['Y'] = df1.index
df1['Y'] = df1['Y'].dt.strftime('%Y')

min_yr = df1.Y.min()
max_yr = df1.Y.max()

# Resample data into 16 day windows of MODIS products
dfs = []
for year in range(int(min_yr), int(max_yr) + 1):
    for win in window_16:

        s = pd.Timestamp(datetime.datetime.strptime(str(year) + win + '0000', '%Y%j%H%M'))

        # MOD conditions
        if (year == 2000 or year == 2004 or year == 2008 or year == 2012 or year == 2016) and (win == '353'):
            jday = str(int(win) + 13)
            e = pd.Timestamp(datetime.datetime.strptime(str(year) + jday.zfill(3) + '1159', '%Y%j%H%M'))
        elif win == '353':
            jday = str(int(win) + 12)
            e = pd.Timestamp(datetime.datetime.strptime(str(year) + jday.zfill(3) + '1159', '%Y%j%H%M'))
```

```python

                # MYD codnitions
                elif (year == 2000 or year == 2004 or year == 2008 or year == 2012 or year == 2016) and (win == '361'):
                    jday = str(int(win) + 5)
                    e = pd.Timestamp(datetime.datetime.strptime(str(year) + jday.zfill(3) + '1159', '%Y%j%H%M'))
                elif win == '361':
                    jday = str(int(win) + 4)
                    e = pd.Timestamp(datetime.datetime.strptime(str(year) + jday.zfill(3) + '1159', '%Y%j%H%M'))

                # 16 day window
                else:
                    jday = str(int(win) + 15)
                    e = pd.Timestamp(datetime.datetime.strptime(str(year) + jday.zfill(3) + '1159', '%Y%j%H%M'))

                dates_mask = (df1['Yjh'] >= s) & (df1['Yjh'] <= e)
                df2 = df1.loc[dates_mask]

                # Check whether the current MODIS 16 Day
                # window has data to compare to the EC
                if df2.empty:
                    pass
                else:
                    df2['period'] = df2.index
                    df2['period'] = df2['period'].dt.strftime('%Y')
                    df2['period'] = df2.period + win

                dfs.append(df2)

        # Check is list are empty: if they are pass
        # other wise resample to 16 Days and save to table
        if not dfs:
            pass
        else:
            df3 = pd.concat(dfs)
            df3.to_csv(out_table + r'\\US_Hourly_' + site + "_" + sat + ".csv")
```

### 6.7 Precip_Regression.py

```python
import datetime
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from scipy import stats

# US Southwest Ameriflux Sites
envs = ['GRA', 'OSH', 'WSA']
amf_dict = {'GRA': ['Aud', 'Dia', 'Scg', 'Seg', 'SRG', 'Wkg'],
            'OSH': ['Scs', 'Scw', 'Ses', 'Whs', 'Wjs'],
            'WSA': ['Fr2', 'Srm', 'Ton', 'Mpj']}

data_path = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Seasons\\'
sats = ['MOD', 'MYD']

GRA = []
OSH = []
WSA = []

# Create figure for linear regression
fig, axes = plt.subplots(nrows=1, ncols=3, constrained_layout=True, sharey=True, figsize=(6, 3))
ax_list = fig.axes

for env, ax in zip(envs, ax_list):
    sites = amf_dict[env]
    for sat in sats:

        print("Loading ENV..." + env)

        # list of all data frames that have dormant season precip
        # and length of growing season
        list_of_df = []
        # Loop through the sites in the current env
        for i in sites:
            if i == 'Scg' or i == 'Scs' or i == 'Scw':
                pass
            else:

                table_g = data_path + r'\Yearly\\US_' + i + '_MOD' + r'.csv'
                table_p = data_path + r'\Precip\\US_' + i + r'.csv'

                dfg = pd.read_csv(table_g, usecols=['period', 'Year', 'length', 'gs'])
                dfp = pd.read_csv(table_p, usecols=['period', 'P'])
                dfg = dfg.set_index(pd.to_datetime(dfg['period'], format="%Y%j"))
                dfp = dfp.set_index(pd.to_datetime(dfp['period'], format="%Y%j"))

                df = pd.merge(dfg, dfp, how='inner', left_index=True, right_index=True)
                df['gl'] = df.Year.astype(str) + df.gs
                df2 = df.groupby('gl').sum()
```

```
                df2['Year'] = df2.index.str.strip().str[:4]
                df2['season'] = df2.index.str.strip().str[-4:]

                df = df2[df2['season'] == 'grow']

                list_of_df.append(df)

        df = pd.concat(list_of_df)
        if env == 'GRA':
            GRA.append(df)
            marker = '^'
            color = "darkorange"
        elif env == 'OSH':
            OSH.append(df)
            marker = 'o'
            color = "lime"
        else:
            WSA.append(df)
            marker = "s"
            color = "magenta"

        df = globals()[env][0]

        # Run statistics
        slope, intercept, r_value, p_value, std_err = stats.linregress(df.P, df.length)
        r_square = np.round(r_value ** 2, 2)
        print(r_square)

        # Create scatter plot w/ titles and labels
        ax.scatter(df.P, df.length, color=color, marker=marker)
        ax.set_title(env)
        ax.set_xlabel('P (mm/yr) \n R$^2$={}'.format(r_square.round(2)))

    ax_list[0].set_ylabel('Growing season \n (Days)')
    plt.savefig(r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Figures\Figures\Figure7.png')
```

### 6.8 PRECIP_GROWSEASONS.PY

```
import datetime
import numpy as np
import pandas as pd

# Path to one drive data folder
path = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Ameriflux\\'

# US Southwest Ameriflux Sites and Coordinates in DD GCW WGS 1984
envs = ['GRA', 'OSH', 'WSA']
amf_dict = {'GRA': ['Aud', 'Dia', 'Scg', 'Seg', 'SRG', 'Wkg'],
            'OSH': ['Scs', 'Scw', 'Ses', 'Whs', 'Wjs'],
            'WSA': ['Fr2', 'Srm', 'Ton', 'Mpj']}

# 16 Day windows based on MODIS products
window_16 = ['001', '017', '033', '049', '065', '081', '097', '113', '129',
             '145', '161', '177', '193', '209', '225',
             '241', '257', '273', '289', '305', '321', '337', '353']

# Loop through all environments - for Precipitation!
fields = ['P']

out_table = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Seasons\Precip\\'
for env in envs:
    sites = amf_dict[env]

    # Loop through the sites in the current env
    for i in sites:
        if i == 'Scg' or i == 'Scs' or i == 'Scw':
            pass
        else:
            fields = ['TIMESTAMP_START', 'P']

            print("Processing Growing/Dormant Season..." + env + " :" + i)

            # Load the original AMF data
            table = path + env + r'\\AMF_US-' + i + '.xlsx'
            df = pd.read_excel(table, usecols=fields, skiprows=2)
            df = df.replace('', np.nan)
            df = df.replace(-9999, np.nan)
            df = df.set_index(pd.to_datetime(df['TIMESTAMP_START'], format="%Y%m%d%H%M"))

            # Convert P reading to mm/day
            (df.P/1800) * 3600 * 24

            # Resample to daily time step
            df = df.resample('D').sum()

            # Add a column that adds its representative
            # Julian Calendar Day of Year (DOY)
            df['Yj'] = df.index
            df['Yj'] = df['Yj'].dt.strftime('%Y%j')

```

```
            # Get min and max Year of data
            df['Y'] = df.index
            df['Y'] = df['Y'].dt.strftime('%Y')

            min_yr = df.Y.min()
            max_yr = df.Y.max()

            df['Yj'] = pd.to_datetime(df['Yj'], format='%Y%j')

            # Resample data into 16 day windows of MODIS products
            dfs = []
            for year in range(int(min_yr), int(max_yr)):
                for count, win in enumerate(window_16):
                    s = datetime.datetime.strptime(str(year) + win, '%Y%j').date()

                    # MODIS avgs the last 12 days of the year
                    # into a 12 day window instead of 16
                    if win == '353':
                        e = datetime.datetime.strptime(str(year) + str(int(win) + 12), '%Y%j')
                    else:
                        e = datetime.datetime.strptime(str(year) + str(int(win) + 15), '%Y%j')

                    dates_mask = (df['Yj'] >= s) & (df['Yj'] <= e)
                    df2 = df.loc[dates_mask]

                    # Check whether the current MODIS 16 Day
                    # window has data to compare to the EC
                    if df2.empty:
                        pass
                    else:
                        df2['period'] = df2.Yj
                        df2['period'] = df2['period'].dt.strftime('%Y')
                        df2['period'] = df2.period + win

                    dfs.append(df2)

            # Check is list are empty: if they are pass other
            # wise resample to 16 Days and save to table
            if not dfs:
                pass
            else:
                df3 = pd.concat(dfs)
                df4 = df3.groupby('period').sum()
                df4.to_csv(out_table + r'\\US_' + i + ".csv")
```

## 6.9 Veg_Activity.py

```python
import datetime
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import matplotlib.dates as mdates
from matplotlib.ticker import FormatStrFormatter


# Path to one drive data folder
path = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Ameriflux\\'

# Path to season results data
season = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Seasons\Yearly\\US_'

# Path to save data
out_table_path = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Seasons\VegAct\\'

# US Southwest Ameriflux Sites
envs = ['GRA', 'OSH', 'WSA']
amf_dict = {'GRA': ['Aud', 'Dia', 'Scg', 'Seg', 'Srg', 'Wkg'],
            'OSH': ['Scs', 'Scw', 'Ses', 'Whs', 'Wjs'],
            'WSA': ['Fr2', 'Srm', 'Ton', 'Mpj']}

# 16 Day windows based on MODIS products
window_16 = ['001', '017', '033', '049', '065', '081', '097', '113', '129',
             '145', '161', '177', '193', '209', '225',
             '241', '257', '273', '289', '305', '321', '337', '353']

seasons = ['grow', 'dorm']

for env in envs:
    grow = []
    dorm = []
    sites = amf_dict[env]

    # Create figure
    fig, axes = plt.subplots(nrows=2, ncols=2, constrained_layout=True, sharex=True, figsize=(6, 6))
    ax_list = fig.axes

    # Loop through the sites in the current env
    for i in sites:
        print("Processing Vegetation Peaks..." + env + " :" + i)

        # Loop through Ameriflux 30 min data
        if i == 'Scg' or i == 'Scs' or i == 'Scw' or i == 'Srg' or i == 'Wkg' or i == 'Whs' \
                or i == 'Srm':
            fields = ['FC', 'LE', 'RH', 'TA', 'TIMESTAMP_START', 'USTAR']
        else:
            fields = ['FC', 'LE', 'VPD', 'TA', 'TIMESTAMP_START', 'USTAR']

```

```
        # Load the original AMF data
        table = path + env + r'\\AMF_US-' + i + '.xlsx'
        df = pd.read_excel(table, usecols=fields, skiprows=2)
        df = df.replace('', np.nan)
        df = df.replace(-9999, np.nan)
        df = df.set_index(pd.to_datetime(df['TIMESTAMP_START'], format="%Y%m%d%H%M"))

        # Get min and max Year of data
        df['Y'] = df.index
        df['Y'] = df['Y'].dt.strftime('%Y')
        min_yr = df.Y.min()
        max_yr = df.Y.max()

        # Add julian days with Years to data
        df['Yj'] = df.index
        df['Yj'] = df['Yj'].dt.strftime('%Y%j')
        df['Yj'] = pd.to_datetime(df['Yj'], format="%Y%j")

        # Add time to readings
        df['HM'] = df.index
        df['HM'] = df['HM'].dt.strftime('%H%M')

        # Calculate the VPD (kPa) for SC sites using Zotarelli et al 2010
        if i == 'Scg' or i == 'Scs' or i == 'Scw' or i == 'Srg' or i == 'Wkg' or i == 'Whs' \
                or i == 'Srm':
            df['etr'] = (0.6108 * np.exp((17.27 * df.TA) / (df.TA + 237.3)))
            df['ea'] = df.etr * (df.RH / 100)
            df['VPD'] = df.etr - df.ea
        else:
            df['VPD'] = df.VPD * 0.1

        # Add MODIS 16 day periods to Ameriflux 30 min data
        df['period'] = np.nan
        for year in range(int(min_yr), int(max_yr) + 1):
            for win in window_16:
                s = pd.Timestamp(datetime.datetime.strptime(str(year) + win, '%Y%j').date())

                # MODIS avgs the last 12 days of the year into a
                # 12 day window instead of 16
                if (year == 2000 or year == 2004 or year == 2008 or year == 2012 or year == 2016) and win == '353':
                    e = pd.Timestamp(datetime.datetime.strptime(str(year) + str(int(win) + 13), '%Y%j').date())
                elif win == '353':
                    e = pd.Timestamp(datetime.datetime.strptime(str(year) + str(int(win) + 12), '%Y%j').date())
                else:
                    e = pd.Timestamp(datetime.datetime.strptime(str(year) + str(int(win) + 15), '%Y%j').date())

                dates_mask = (df['Yj'] >= s) & (df['Yj'] <= e)
                df.loc[dates_mask, 'period'] = str(year) + win

                # Get seasonal data from Season tables
```

```
                sea_table = pd.read_csv(season + i + ".csv", usecols=['period', 'gs'])
                sea_table['period'] = sea_table['period'].astype(str)

                df1 = pd.merge(df, sea_table, on='period', how='inner')
                df1 = df1.set_index(pd.to_datetime(df1['TIMESTAMP_START'], format="%Y%m%d%H%M"))

                # Filter data using parameters of Methodology
                # Keep data above Friction Velocity of 0.1 m/s
                df1 = df1[df1['USTAR'] > 0.1]

                # Convert FC to gC/30 hr
                df1['gC'] = (df1.FC * 1E-6 * 44.01 * 3600)

                # Keep only data of the CO2 Flux, VPD, TA that is between
                # 2 Standard Deviations within the mean
                df1 = df1[((df1.FC - df1.FC.mean()) / df1.FC.std()).abs() < 2]
                df1 = df1[((df1.VPD - df1.VPD.mean()) / df1.VPD.std()).abs() < 2]
                df1 = df1[((df1.TA - df1.TA.mean()) / df1.TA.std()).abs() < 2]

                # Filter for grow and dormant season
                for x in seasons:
                    df2 = df1[df1['gs'] == x]
                    df2 = df2.groupby('HM').mean()
                    globals()[x].append(df2)

            # Merge data to get avg curve
            dfg = pd.concat(grow, axis=0, sort=True)
            dfg = dfg.groupby(dfg.index).mean()
            dfd = pd.concat(dorm, axis=0, sort=True)
            dfd = dfd.groupby(dfd.index).mean()

            # Save CSV file
            dfd.to_csv(out_table_path + 'US_dorm_' + env + '.csv')
            dfg.to_csv(out_table_path + 'US_grow_' + env + '.csv')
```

## 6.10 Veg_Act_Fig.py

```python
import matplotlib.pyplot as plt
import pandas as pd
import matplotlib.dates as mdates
from matplotlib.ticker import FormatStrFormatter

# Path to saved data
table_path = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Seasons\VegAct\\'

# US Southwest Ameriflux Sites Env
envs = ['GRA', 'OSH', 'WSA']

# Create figure
fig, axes = plt.subplots(nrows=3, ncols=4, constrained_layout=True, sharex=True, figsize=(8.5, 5.5))

# Field to load in pandas
fields = ['TA', 'gC', 'VPD', 'HM']

cols = [('$\it{FC}$ (gCO$_2$ m$^2$ hr)'), ('$\it{λLE}$$_c$ (W/m$^2$)'), ('VPD (kPa)'), ('TA (°C)')]
# Set column names: only top ones!
for ax, col in zip(axes[0], cols):
    ax.set_title(col, fontsize=9)

rows = ['{}'.format(row) for row in ['GRA', 'OSH', 'WSA']]
# Set row names
for ax, row in zip(axes[:,0], rows):
    ax.set_ylabel(row, rotation=0, fontsize=10, labelpad=15)

for env, axe in zip(envs, axes):
    print("Loading..." + env)
    # Load data
    dfd = pd.read_csv(table_path + 'US_dorm_' + env + '.csv', usecols=fields)
    dfg = pd.read_csv(table_path + 'US_grow_' + env + '.csv', usecols=fields)
    dld = pd.read_csv(table_path + 'US_dorm_LEc_' + env + '.csv', usecols=['HM', 'LE'])
    dlg = pd.read_csv(table_path + 'US_grow_LEc_' + env + '.csv', usecols=['HM', 'LE'])

    # Add leading zeros
    dfd.HM = dfd.HM.astype(str).str.zfill(4)
    dfg.HM = dfg.HM.astype(str).str.zfill(4)
    dld.HM = dld.HM.astype(str).str.zfill(4)
    dlg.HM = dlg.HM.astype(str).str.zfill(4)

    # Set index as time
    dfd = dfd.set_index(pd.to_datetime(dfd['HM'], format="%H%M"))
    dfg = dfg.set_index(pd.to_datetime(dfg['HM'], format="%H%M"))
    dld = dld.set_index(pd.to_datetime(dld['HM'], format="%H%M"))
    dlg = dlg.set_index(pd.to_datetime(dlg['HM'], format="%H%M"))
    variabs = ['gC', 'LE', 'TA', 'VPD']

    for x in variabs:
        if x == 'LE':
```

```
            # Print max results:
            print()
            print("The peak of LEC (dorm) " + x + " time is... ")
            print(dld[dld[x] == dld[x].max()])
            print("The peak of LEC (grow) " + x + " time is... ")
            print(dlg[dlg[x] == dlg[x].max()])
        elif x == 'VPD':
            print()
            print("The min of VDP (dorm) " + x + " time is... ")
            print(dfd[dfd[x] == dfd[x].min()])
            print("The min of VDP (grow) " + x + " time is... ")
            print(dfg[dfg[x] == dfg[x].min()])
            # Print max results:
        elif x == 'gC':
            print()
            print("The min of FC (dorm) " + x + " time is... ")
            print(dfd[dfd[x] == dfd[x].min()])
            print("The min of FC (grow) " + x + " time is... ")
            print(dfg[dfg[x] == dfg[x].min()])
            # Print max results:
        else:
            print()
            print("The max T(dorm) " + x + " time is... ")
            print(dfd[dfd[x] == dfd[x].max()])
            print("The max T(grow) " + x + " time is... ")
            print(dfg[dfg[x] == dfg[x].max()])

    # Plot the data
    for count, ax in enumerate(axe):
        # Set # of sig figures
        ax.yaxis.set_major_formatter(FormatStrFormatter('%.2f'))
        ax.xaxis.set_tick_params(labelsize=7)
        ax.yaxis.set_tick_params(labelsize=7)

        if count == 0:
            ax.plot(dfd.index, dfd.gC, color='red', linewidth=1)
            ax.plot(dfg.index, dfg.gC, color='green', linewidth=1)
        elif count == 1:
            ax.plot(dld.index, dld.LE, color='red', linewidth=1)
            ax.plot(dlg.index, dlg.LE, color='green', linewidth=1)
        elif count == 2:
            ax.plot(dfd.index, dfd.VPD, color='red', linewidth=1)
            ax.plot(dfg.index, dfg.VPD, color='green', linewidth=1)
        else:
            ax.plot(dfd.index, dfd.TA, color='red', linewidth=1)
            ax.plot(dfg.index, dfg.TA, color='green', linewidth=1)

    # Time format for ticks
    hours = mdates.HourLocator(byhour=range(0, 24, 6))
           sub_hour = mdates.HourLocator(interval=1)
           timeFmt = mdates.DateFormatter('%I%p')

           # Format ticks
           ax.xaxis.set_major_locator(hours)
           ax.xaxis.set_major_formatter(timeFmt)
           ax.xaxis.set_minor_locator(sub_hour)

```

```
        plt.savefig(r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Figures\Figures\Figure5.png', dpi=600)
```

## 6.11 VEG_ACT_LEC.PY

```
import datetime
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Path to one drive data folder
path = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Ameriflux\\'

# Path to season results data
season = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Seasons\Yearly\\US_'

# Path to save data
out_table_path = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Seasons\VegAct\\'

# US Southwest Ameriflux Sites
envs = ['GRA', 'OSH', 'WSA']
amf_dict = {'GRA': ['Aud', 'Dia', 'Scg', 'Seg', 'Srg', 'Wkg'],
            'OSH': ['Scs', 'Scw', 'Ses', 'Whs', 'Wjs'],
            'WSA': ['Fr2', 'Srm', 'Ton', 'Mpj']}

# 16 Day windows based on MODIS products
window_16 = ['001', '017', '033', '049', '065', '081', '097', '113', '129',
             '145', '161', '177', '193', '209', '225',
             '241', '257', '273', '289', '305', '321', '337', '353']

seasons = ['grow', 'dorm']
for env in envs:
    grow = []
    dorm = []
    sites = amf_dict[env]

    # Create figure
    fig, axes = plt.subplots(nrows=2, ncols=2, constrained_layout=True, sharex=True, figsize=(6, 6))
    ax_list = fig.axes

    # Loop through the sites in the current env
    for i in sites:
        print("Processing Vegetation Peaks..." + env + " :" + i)

        # Load the 1 Hr sampled AMF data
        table = path + env + r'\\US_1HrAvg_' + i + '.csv'
        df = pd.read_csv(table)
        df = df.replace('', np.nan)
        df = df.replace(-9999, np.nan)
        df = df.set_index(pd.to_datetime(df['TIMESTAMP_START'], format="%Y-%m-%d %H:%M"))
```

```python

        # Get min and max Year of data
        df['Y'] = df.index
        df['Y'] = df['Y'].dt.strftime('%Y')
        min_yr = df.Y.min()
        max_yr = df.Y.max()

        # Add julian days with Years to data
        df['Yj'] = df.index
        df['Yj'] = df['Yj'].dt.strftime('%Y%j')
        df['Yj'] = pd.to_datetime(df['Yj'], format="%Y%j")

        # Add time to readings
        df['HM'] = df.index
        df['HM'] = df['HM'].dt.strftime('%H%M')

        # Add MODIS 16 day periods to Ameriflux 30 min data
        df['period'] = np.nan
        for year in range(int(min_yr), int(max_yr) + 1):
            for win in window_16:
                s = pd.Timestamp(datetime.datetime.strptime(str(year) + win, '%Y%j').date())

                # MODIS avgs the last 12 days of the year
                # into a 12 day window instead of 16
                if (year == 2000 or year == 2004 or year == 2008 or year == 2012 or year == 2016) and win == '353':
                    e = pd.Timestamp(datetime.datetime.strptime(str(year) + str(int(win) + 13), '%Y%j').date())
                elif win == '353':
                    e = pd.Timestamp(datetime.datetime.strptime(str(year) + str(int(win) + 12), '%Y%j').date())
                else:
                    e = pd.Timestamp(datetime.datetime.strptime(str(year) + str(int(win) + 15), '%Y%j').date())

                dates_mask = (df['Yj'] >= s) & (df['Yj'] <= e)
                df.loc[dates_mask, 'period'] = str(year) + win

        # Get seasonal data from Season tables
        sea_table = pd.read_csv(season + i + ".csv", usecols=['period', 'gs'])
        sea_table['period'] = sea_table['period'].astype(str)

        df1 = pd.merge(df, sea_table, on='period', how='inner')
        df1 = df1.set_index(pd.to_datetime(df1['TIMESTAMP_START'], format="%Y-%m-%d %H:%M"))

        # Filter for grow and dormant season
        for x in seasons:
            df2 = df1[df1['gs'] == x]
            df2 = df2.groupby('HM').mean()
            globals()[x].append(df2)

    # Merge data to get avg curve
    dfg = pd.concat(grow, axis=0, sort=True)
    dfg = dfg.groupby(dfg.index).mean()
    dfd = pd.concat(dorm, axis=0, sort=True)
```

```python
    dfd = dfd.groupby(dfd.index).mean()

    # Save CSV file
            dfd.to_csv(out_table_path + 'US_dorm_LEc_' + env + '.csv')
            dfg.to_csv(out_table_path + 'US_grow_LEc_' + env + '.csv')
```

## 6.12 NDVI_SC.PY

```python
import datetime
import numpy as np
import pandas as pd

# Path to one drive data folder
path = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Ameriflux\\'

# Path to season results data
season = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Seasons\Yearly\\US_'

# Path to save data
out_table_path = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Seasons\VegAct_sc\\'

# US Southwest Ameriflux Sites
envs = ['GRA', 'OSH']
amf_dict = {'GRA': ['Scg'],
            'OSH': ['Scs', 'Scw']}

# These will be the fields used at each site
fields = ['TIMESTAMP_START', 'USTAR', 'SWC', 'NDVI', 'LE', 'NETRAD', 'SW_IN', 'TA', 'TIMESTAMP_END', 'FC', 'RH']

# 16 Day windows based on MODIS products
window_16 = ['001', '017', '033', '049', '065', '081', '097', '113', '129', '145', '161', '177', '193', '209', '225',
             '241', '257', '273', '289', '305', '321', '337', '353']

seasons = ['grow', 'dorm']

for env in envs:
    grow = []
    dorm = []
    sites = amf_dict[env]

    for i in sites:
        print("Processing Vegetation..." + env + " :" + i)
        # Load the original AMF data
        table = path + env + r'\\US_1HrAvg_SOCA_' + i + '.csv'
        df = pd.read_csv(table, usecols=fields)
        df = df.replace('', np.nan)
        df = df.replace(-9999, np.nan)
```

```
        df = df.set_index(pd.to_datetime(df['TIMESTAMP_START'], format="%Y-%m-%d %H:%M"))

        # Calculate the VPD
        df['etr'] = (0.6108 * np.exp((17.27 * df.TA) / (df.TA + 237.3)))
        df['ea'] = df.etr * (df.RH / 100)
        df['VPD'] = df.etr - df.ea

        # Get min and max Year of data
        df['Y'] = df.index
        df['Y'] = df['Y'].dt.strftime('%Y')
        min_yr = df.Y.min()
        max_yr = df.Y.max()

        # Add julian days with Years to data
        df['Yj'] = df.index
        df['Yj'] = df['Yj'].dt.strftime('%Y%j')
        df['Yj'] = pd.to_datetime(df['Yj'], format="%Y%j")

        # Add time to readings
        df['HM'] = df.index
        df['HM'] = df['HM'].dt.strftime('%H%M')

        # Add MODIS 16 day periods to Ameriflux 30 min data
        df['period'] = np.nan
        for year in range(int(min_yr), int(max_yr) + 1):
            for win in window_16:
                s = pd.Timestamp(datetime.datetime.strptime(str(year) + win, '%Y%j').date())

                # MODIS avgs the last 12 days of the year into
                # a 12 day window instead of 16
                if (year == 2000 or year == 2004 or year == 2008 or year == 2012 or year == 2016) and win == '353':
                    e = pd.Timestamp(datetime.datetime.strptime(str(year) + str(int(win) + 13), '%Y%j').date())
                elif win == '353':
                    e = pd.Timestamp(datetime.datetime.strptime(str(year) + str(int(win) + 12), '%Y%j').date())
                else:
                    e = pd.Timestamp(datetime.datetime.strptime(str(year) + str(int(win) + 15), '%Y%j').date())

                dates_mask = (df['Yj'] >= s) & (df['Yj'] <= e)
                df.loc[dates_mask, 'period'] = str(year) + win

        # Get seasonal data from Season tables
        sea_table = pd.read_csv(season + i + ".csv", usecols=['period', 'gs'])
        sea_table['period'] = sea_table['period'].astype(str)

        df1 = pd.merge(df, sea_table, on='period', how='inner')
        df1 = df1.set_index(pd.to_datetime(df1['TIMESTAMP_START'], format="%Y-%m-%d %H:%M"))

        # Convert FC to gC/30 hr
        df1['gC'] = (df1.FC * 1E-6 * 44.01 * 3600)

```

```
        # Keep only data of the CO2 Flux, VPD, TA that is
        # between 2 Standard Deviations within the mean
        df1 = df1[((df1.FC - df1.FC.mean()) / df1.FC.std()).abs() < 2]
        df1 = df1[((df1.VPD - df1.VPD.mean()) / df1.VPD.std()).abs() < 2]
        df1 = df1[((df1.TA - df1.TA.mean()) / df1.TA.std()).abs() < 2]

        # Filter for grow and dormant season
        for x in seasons:
            df2 = df1[df1['gs'] == x]
            # Save CSV file
                df2.to_csv(out_table_path + 'US_' + x + '_' + i + '.csv')


```

## 6.13 NDVI_CURVE.PY

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from matplotlib.ticker import FormatStrFormatter
import matplotlib.dates as mdates

# Path to save data
data_path = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Seasons\VegAct_sc\\US_grow_'

date_input = input("Enter a today's date in the following format (YYYY-MM-DD): ")
# SOCA sites
sites = ['Scg', 'Scs', 'Scw']
fig, axes = plt.subplots(nrows=1, ncols=3, constrained_layout=True, sharex=True, sharey=True, figsize=(6, 3))
fields = ['TIMESTAMP_START', 'NDVI']

# For loop for sites
for i, ax in zip(sites, axes):
    # Define the env being analyzed
    if i == 'Scg':
        env = 'GRA'
    else:
        env = 'OSH'
    print("\t Loading... " + i)
    df = pd.read_csv(data_path + i + '.csv', usecols=fields)
    df = df.set_index(pd.to_datetime(df['TIMESTAMP_START'], format="%Y-%m-%d %H:%M"))
    df = df.drop(columns=['TIMESTAMP_START'])
    df['Time'] = df.index
    df['Time'] = df['Time'].dt.strftime('%H:%M')

    # Create diurnal curves
    df = df.groupby('Time').mean()
    df.index = pd.to_datetime(df.index.astype(str))

    # Import Satellite Data
```

```python
    mod_ndvi = pd.read_csv(r'E:\C4_ET\Data\Tables\NDVI\\' + i + '_MOD13Q1_NDVI.csv')
    myd_ndvu = pd.read_csv(r'E:\C4_ET\Data\Tables\NDVI\\' + i + '_MYD13Q1_NDVI.csv')

    # Get mean of NDVI values
    terra = mod_ndvi['NDVI'].mean()
    aqua = myd_ndvu['NDVI'].mean()

    # Ovrpass times
    terra_ovr = pd.Timestamp(date_input + ' 11:00:00')
    aqua_ovr = pd.Timestamp(date_input + ' 13:00:00')

    # Plot the data
    ax.plot(df.index, df.NDVI, 'r', linewidth=1, label='NDVI')
    ax.scatter(terra_ovr, terra, marker='^', color='g', label='Terra')
    ax.scatter(aqua_ovr, aqua, marker='s', color='b', label='Aqua')
    ax.yaxis.set_major_formatter(FormatStrFormatter('%.2f'))

    # Set x labels
    # Time format for ticks
    hours = mdates.HourLocator(byhour=range(0, 24, 6))
    sub_hour = mdates.HourLocator(interval=1)
    timeFmt = mdates.DateFormatter('%I%p')

    # Format ticks
    ax.xaxis.set_major_locator(hours)
    ax.xaxis.set_major_formatter(timeFmt)
    ax.xaxis.set_minor_locator(sub_hour)

    # Set Axis to 0500AM - 0700PM
    min = pd.Timestamp(date_input + ' 05:00:00')
    max = pd.Timestamp(date_input + ' 18:00:00')
    datemin = np.datetime64(min, 'h') - np.timedelta64(1, 'h')
    datemax = np.datetime64(max, 'h') + np.timedelta64(1, 'h')
    ax.set_xlim(datemin, datemax)

    # Set titles
    ax.set_title(i + ' ({})'.format(env))

    # Find the over/under estimation of NDVI
    g_terra = df.loc[terra_ovr][0]
    g_aqua = df.loc[aqua_ovr][0]
    mean_g_ndvi = (g_terra + g_aqua)/2
    mean_sat_ndvi = (terra + aqua) / 2
    print(i)
    print('Mean MODIS NDVI: ' + str(mean_sat_ndvi.round(3)))
    print('Mean ground NDVI: ' + str(mean_g_ndvi.round(3)))
    print('Terra NDVI: ' + str(terra.round(3)))
    print('Aqua NDVI: ' + str(aqua.round(3)))
    print('Terra NDVI Ground: ' + str(g_terra.round(3)))
    print('Aqua NDVI Ground: ' + str(g_aqua.round(3)))
    print("______________________________________________")

# Set Y-label
axes[0].set_ylabel('NDVI')
axes[2].legend()
```

```
plt.savefig(r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Figures\Figures\NDVI_diurnal.png')
```

## 6.14 NDVI_REG.PY

```
import numpy as np
import pandas as pd
from scipy import stats
from sklearn.metrics import mean_squared_error
from math import sqrt

# Path to season results data
data_path = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Seasons\VegAct_sc\\US_'

# Path to save data
out_table_path = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Seasons\VegAct_sc\\'

# US Southwest Ameriflux Sites
envs = ['GRA', 'OSH']
amf_dict = {'GRA': ['Scg'],
            'OSH': ['Scs', 'Scw']}
seasons = ['grow', 'dorm']

for env in envs:
    sites = amf_dict[env]
    for i in sites:
        for season in seasons:
            print("Site: " + i + "....Season: " + season)
            table = data_path + season + '_' + i + '.csv'
            df = pd.read_csv(table)
            df = df.replace('', np.nan)
            df = df.dropna()

            # Run statistics
            slope_fc, interceptf, fc_r, fp_value, std_err = stats.linregress(df.NDVI, df.FC)
            slope_VPD, interceptv, vpd_r, vp_value, std_err = stats.linregress(df.NDVI, df.VPD)
            slope_LE, interceptl, le_r, lp_value, std_err = stats.linregress(df.NDVI, df.LE)

            # Make lines and RMSE
            def clim(a, m,b):
                return a * m + b

            params = ['gC', 'LE', 'VPD']
            x = df['NDVI'].values

            for param in params:
                y = df[param].values

```

```
            if param == 'FC':
                eq = np.polyfit(df.NDVI, df.gC, 1)
            elif param == 'VPD':
                eq = np.polyfit(df.NDVI, df.VPD, 1)
            else:
                eq = np.polyfit(df.NDVI, df.LE, 1)

            y_model = clim(x, eq[0], eq[1])
            rmse = sqrt(mean_squared_error(y, y_model))
            rmse = np.round(rmse, 4)
            print('\t RMSE ' + param + ": " + str(rmse))

        # Round r square values
        fc_r = np.round(fc_r ** 2, 2)
        le_r = np.round(le_r ** 2, 2)
        vpd_r = np.round(vpd_r ** 2, 2)

        # Print P values
        if fp_value < 0.05:
            fp_val = 'p < 0.05'
        else:
            fp_val = 'p > 0.05'

        if vp_value < 0.05:
            vp_val = 'p < 0.05'
        else:
            vp_val = 'p > 0.05'

        if lp_value < 0.05:
            lp_val = 'p < 0.05'
        else:
            lp_val = 'p > 0.05'

        # Round slope
        fcs = np.round(slope_fc, 2)
        les = np.round(slope_LE, 2)
        vpds = np.round(slope_VPD, 2)

        # Round intercept
        fcb = np.round(interceptf, 2)
        leb = np.round(interceptl, 2)
        vpdb = np.round(interceptv, 2)

        print("\t FC r^2 = " + str(fc_r), 'FC m = ' + str(fcs), 'FC b =' + str(fcb))
        print("\t LE r^2 = " + str(le_r), 'LE m = ' + str(les), 'LE b =' + str(leb))
        print("\t VPD r^2 = " + str(vpd_r), 'VPD m = ' + str(vpds), 'VPD b =' + str(vpdb))
        print("\t FC pval = " + str(fp_val), "VPD pval = " + str(vp_val), 'l pval =  ' + str(lp_val))
```

## 6.15 Kc_Reg_Day.py

```python
import pandas as pd

# Path to one drive data folder
eto_table = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\ETo\\'

# Path to one drive data folder
path = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Ameriflux\\'

# EVI path
evi_path = r'E:\C4_ET\Data\Tables\EVI\\'

# Path to save tables
out_table = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Reg_Day\\'

# Path to Growing Season data
season_path = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Seasons\Yearly\\'

# Environments analysed (Table 1)
envs = ['GRA', 'OSH', 'WSA']

# US Southwest Ameriflux Sites (Table 2)
amf_dict = {'GRA': ['Aud', 'Dia', 'Scg', 'Seg', 'Srg', 'Wkg'],
            'OSH': ['Scs', 'Scw', 'Ses', 'Whs', 'Wjs'],
            'WSA': ['Fr2', 'Srm', 'Ton', 'Mpj']}

sats = ['MOD', 'MYD']
# Fields to use
fields = ['TIMESTAMP_START', 'FC', 'LE']
for env in envs:
    for sat in sats:

        dfs = []
        grow = []
        dorm = []
        sites = amf_dict[env]
        for i in sites:
            # ETo Daily Data; ETo data was processed with filtered 1 hour data
            df = pd.read_csv(eto_table + "US_Daily_" + i + '_' + sat + ".csv", usecols=['Kc', 'period'])
            df = df.set_index(pd.to_datetime(df['period'], format="%Y%j"))

            # Load MODIS data
            evi = pd.read_csv(evi_path + i + '_' + sat + '13Q1_EVI.csv')
            evi_cols = ['period', sat + '_EVI']
            evi.columns = evi_cols
            evi = evi.set_index(pd.to_datetime(evi['period'], format="%Y-%m-%d %H:%M"))

            # Merge MOD data with dataframe using the MOD period
            df1 = pd.merge(df, evi, how='inner', left_index=True, right_index=True)
```

```
            df1 = df1.drop(columns=['period_x', 'period_y'])

            # Add growing/dormant season
            season = pd.read_csv(season_path + 'US_' + i + '_' + sat + '.csv', usecols=['period', 'gs'])
            season = season.set_index(pd.to_datetime(season['period'], format="%Y%j"))
            df2 = pd.merge(df1, season, how='inner', left_index=True, right_index=True)

            # Append data to env list
            dfs.append(df2)

        # Merge all data for ENV
        df3 = pd.concat(dfs)
        df3.to_csv(out_table + env + '_' + sat + '.csv')
```

## 6.16 KC_REG_OVR.PY

```
import pandas as pd

# Path to one drive data folder
eto_table = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\ETo\\'

# Path to one drive data folder
path = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Ameriflux\\'

# Time path
time_path = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Time\\'

# EVI path
evi_path = r'E:\C4_ET\Data\Tables\EVI\\'

# Path to save tables
out_table = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Reg_Ovr\\'

# Path to Growing Season data
season_path = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Seasons\Yearly\\'

# Environments analysed (Table 1)
envs = ['GRA', 'OSH', 'WSA']

# US Southwest Ameriflux Sites (Table 2)
amf_dict = {'GRA': ['Aud', 'Dia', 'Scg', 'Seg', 'Srg', 'Wkg'],
            'OSH': ['Scs', 'Scw', 'Ses', 'Whs', 'Wjs'],
            'WSA': ['Fr2', 'Srm', 'Ton', 'Mpj']}

# Time zones for UTC offsets
tz_utc = {'Aud': 7,
          'Dia': 8,
```

```
          'Scg': 8,
          'Seg': 7,
          'Srg': 7,
          'Wkg': 7,
          'Mpj': 7,
          'Scs': 8,
          'Scw': 8,
          'Ses': 7,
          'Whs': 7,
          'Wjs': 7,
          'Ton': 8,
          'Srm': 7,
          'Fr2': 6}

sats = ['MOD', 'MYD']
# Fields to use
fields = ['TIMESTAMP_START', 'FC', 'LE']
for env in envs:
    for sat in sats:

        dfs = []
        grow = []
        dorm = []
        sites = amf_dict[env]
        for i in sites:
            # Filtered 1 Hr Ameriflux Data
            df_le = pd.read_csv(path + env + r'\\US_1HrAvg_' + i + ".csv", usecols=fields)
            df_le = df_le.set_index(pd.to_datetime(df_le['TIMESTAMP_START'], format="%Y-%m-%d %H:%M"))

            # ETo hourly data
            df_eto = pd.read_csv(eto_table + "US_Hourly_" + i + '_' + sat + ".csv", usecols=['TIMESTAMP_START', 'Kc', 'period'])
            df_eto = df_eto.set_index(pd.to_datetime(df_eto['TIMESTAMP_START'], format="%Y-%m-%d %H:%M"))

            # Merge ETo data to filtered 1 hr data
            df = pd.merge(df_le, df_eto, how='inner', left_index=True, right_index=True)
            df = df.drop(columns=['TIMESTAMP_START_x', 'TIMESTAMP_START_y'])

            # Add a column with %Y%H%MODISPERIOD and Hour
            df['period'] = df['period'].astype(str)
            df['H'] = df.index
            df['H'] = df['H'].dt.strftime('%H')

            # Load MODIS data
            evi = pd.read_csv(evi_path + i + '_' + sat + '13Q1_EVI.csv')
            evi_cols = ['Date', sat + '_EVI']
            evi.columns = evi_cols
            evi = evi.set_index(pd.to_datetime(evi['Date'], format="%Y-%m-%d"))
            evi = evi.drop(columns='Date')

            # Load MOD Times in UTC
            time = pd.read_csv(time_path + i + '_' + sat + '11A2.csv')
```

```
            time = time.set_index(pd.to_datetime(time['period'], format="%Y%j"))

            # Merge MOD data
            df_mod = pd.merge(evi, time, how='inner', left_index=True, right_index=True)
            df_mod = df_mod.reset_index()
            df_mod['period'] = df_mod['period'].astype(str)

            # Merge MOD data with dataframe using the MOD period
            df1 = pd.merge(df, df_mod, on=['period'])
            df1 = df1.round({'UTC_Time': 0})
            df1['Time_local'] = df1.UTC_Time - tz_utc[i]

            # Merge MOd data with Ameriflux
            df1['H'] = df1['H'].astype(int)
            df2 = df1[df1.H == df1.Time_local]

            # Grouby the mod period
                  df3 = df2.groupby(df2.period).mean()

                  # Add growing/dormant season
                  season = pd.read_csv(season_path + 'US_' + i + '_' + sat + '.csv',  usecols=['period', 'gs'])
                  season['period'] = season['period'].astype(str)
                  df4 = pd.merge(df3, season, on=['period'])

                  # Append data to env list
                  dfs.append(df4)

              # Merge all data for ENV
              df5 = pd.concat(dfs)
              df5.to_csv(out_table + env + '_' + sat + '.csv')
```

## 6.17 16D_RESAMPLE.PY

```
import datetime
import numpy as np
import pandas as pd

# Path to one Time data
path = r'E:\C4_ET\Data\Tables\Time\\'

# Path to save data
out_table_path = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Time\\'

# US Southwest Ameriflux Sites
envs = ['GRA', 'OSH', 'WSA']
amf_dict = {'GRA': ['Aud', 'Dia', 'Scg', 'Seg', 'Srg', 'Wkg'],
            'OSH': ['Scs', 'Scw', 'Ses', 'Whs', 'Wjs'],
            'WSA': ['Fr2', 'Srm', 'Ton', 'Mpj']}

```

```python
# 16 Day windows based on MODIS products
window_mod = ['001', '017', '033', '049', '065', '081', '097', '113', '129',
              '145', '161', '177', '193', '209', '225',
              '241', '257', '273', '289', '305', '321', '337', '353']

window_myd = ['009', '025', '041', '057', '073', '089', '105', '121', '137',
              '153', '169', '185', '201', '217', '233',
              '249', '265', '281', '297', '313', '329', '345', '361']

mod_prods = ['MOD11A2', 'MYD11A2']

for mod in mod_prods:
    if mod == 'MOD11A2':
        window_16 = window_mod
    else:
        window_16 = window_myd

    for env in envs:
        sites = amf_dict[env]

        # Loop through the sites in the current env
        for i in sites:
            print("Processing Satellite Overpass..." + env + " :" + i)

            # Load the time data
            table = path + i + "_" + mod + ".csv"
            df = pd.read_csv(table)
            if mod == 'MOD11A2':
                mod_columns = ['Date', 'Day_View', 'UTC_Time']
            else:
                mod_columns = ['Date', 'Day_View', 'UTC_Time']
            df.columns = mod_columns
            df = df.set_index(pd.to_datetime(df['Date'], format="%Y-%m-%d"))
            df = df.drop(columns=['Date', 'Day_View'])

            # Get min and max Year of data
            df['Y'] = df.index
            df['Y'] = df['Y'].dt.strftime('%Y')
            min_yr = df.Y.min()
            max_yr = df.Y.max()

            # Add julian days with Years to data
            df['Yj'] = df.index
            df['Yj'] = df['Yj'].dt.strftime('%Y%j')
            df['Yj'] = pd.to_datetime(df['Yj'], format="%Y%j")

            # Add MODIS 16 day periods to Ameriflux 30 min data
            df['period'] = np.nan
            for year in range(int(min_yr), int(max_yr) + 1):
                for win in window_16:
                    s = pd.Timestamp(datetime.datetime.strptime(str(year) + win, '%Y%j').date())

                    # MOD conditions
                    if (year == 2000 or year == 2004 or year == 2008 or year == 2012 or year == 2016) and (win == '353'):
                        jday = str(int(win) + 13)
```

```
                    e = pd.Timestamp(datetime.datetime.strptime(str(year) + jday.zfill(3) + '1159', '%Y%j%H%M'))
                elif win == '353':
                    jday = str(int(win) + 12)
                    e = pd.Timestamp(datetime.datetime.strptime(str(year) + jday.zfill(3) + '1159', '%Y%j%H%M'))

                # MYD codnitions
                elif (year == 2000 or year == 2004 or year == 2008 or year == 2012 or year == 2016) and (win == '361'):
                    jday = str(int(win) + 5)
                    e = pd.Timestamp(datetime.datetime.strptime(str(year) + jday.zfill(3) + '1159', '%Y%j%H%M'))
                elif win == '361':
                    jday = str(int(win) + 4)
                    e = pd.Timestamp(datetime.datetime.strptime(str(year) + jday.zfill(3) + '1159', '%Y%j%H%M'))

                # 16 day window
                else:
                    jday = str(int(win) + 15)
                    e = pd.Timestamp(datetime.datetime.strptime(str(year) + jday.zfill(3) + '1159', '%Y%j%H%M'))

                dates_mask = (df['Yj'] >= s) & (df['Yj'] <= e)
                df.loc[dates_mask, 'period'] = str(year) + win

        # Take mean of satellite overass
        df = df.groupby(df.period).mean()

        # Save CSV file
        df.to_csv(out_table_path + i + "_" + mod + ".csv")
```

### 6.18 Kc_Reg_Fig.py

```
import matplotlib.pyplot as plt
import pandas as pd
from scipy import stats
import numpy as np
from sklearn.metrics import mean_squared_error
from math import sqrt

# Path to saved data
table_path = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Reg_Day\\'

# Create figure
fig, axes = plt.subplots(nrows=4, ncols=3, constrained_layout=True, sharex=True, sharey=True, figsize=(8.5, 4.02))

cols = [('GRA'), ('OSH'), ('WSA')]
# Set column names: only top ones!
for ax, col in zip(axes[0], cols):
    ax.set_title(col, fontsize=9)

```

```
# Plot the data
for rcnt, row in enumerate(axes):
    # Set the satellite being used
    if rcnt == 0 or rcnt == 1:
        sat = 'MOD'
    else:
        sat = 'MYD'

    # Set the dorm/growing season
    if rcnt == 0 or rcnt == 2:
        ssn = 'grow'
        figm = 'G'
    else:
        ssn = 'dorm'
        figm = 'D'

    # Plot the data by row/col in figure
    for count, ax in enumerate(row):
        if count == 0:
            table = table_path + 'GRA_' + sat + '.csv'
            marker = '^'
            color = "darkorange"
            ax.set_ylabel('K$_c$', fontsize=10)
        elif count == 1:
            table = table_path + 'OSH_' + sat + '.csv'
            marker = 'o'
            color = "lime"
        else:
            table = table_path + 'WSA_' + sat + '.csv'
            marker = "s"
            color = "magenta"

        if rcnt == 3 and (count == count):
            ax.set_xlabel('EVI', fontsize=8)

        df = pd.read_csv(table)
        df = df[df['gs'] == ssn]
        name = sat + '_EVI'
        # Run statistics
        slope, intercept, r_value, p_value, std_err = stats.linregress(df[name], df.Kc)
        r_square = np.round(r_value ** 2, 2)
        if ssn == 'grow':
            print(np.round(slope, 4))
            print(np.round(intercept,4))
        else:
            pass

        if p_value < 0.05:
            p_val = 'p < 0.05'
        else:
            p_val = 'p > 0.05'

        # Calculate RMSE
        def clim(a, m, b):
            return a * m + b
```

```
        x = df[name].values
        y = df['Kc'].values
        y_model = clim(x, slope, intercept)
        rmse = sqrt(mean_squared_error(y, y_model))
        rmse = np.round(rmse, 4)
        ax.scatter(df[name], df.Kc, color=color, marker=marker)

        # Set labels only on the left and right
        ax.text(0.4375, 1.120, 'R$^2$={}\n{}'.format(str.format('{0:.2f}', r_square),  p_val), ha='center', fontsize=8, bbox=dict(facecolor='white'))
        ax.text(0.3, 1.65, sat + ' ({})'.format(figm), ha='center', fontsize=8)

        ax.text(0.3, 1.30, 'RMSE=±{}'.format(str.format('{0:.2f}', rmse)), ha='center', fontsize=8)
plt.savefig(r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Figures\Figure6_ovr.png', dpi=600)
```

## 6.19 Kc_ExpReg.py

```
import pandas as pd
import numpy as np
from scipy.optimize import curve_fit
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
from math import sqrt

# Path to saved data
table_path = r'C:\Users\laptop2\OneDrive - The University of Texas at Austin\Thesis\Data\Reg_Day\\'

envs = ['GRA', 'OSH', 'WSA']
for env in envs:
    table = table_path + env + "_MYD.csv"
    df = pd.read_csv(table)
    df = df[df['gs'] == 'grow']

    def kc(evi, a,b,c):
        return a * (1-np.exp(-b * evi)) - c

    x = df['MYD_EVI'].values
    y = df['Kc'].values
    c,cov = curve_fit(kc, x, y, maxfev=2000)

    y_kc = kc(x, c[0], c[1], c[2])

    coeffs = 'kc = {} * (1 -exp (-{} * evi)) - {}'.format(c[0].round(2), c[1].round(5), c[2].round(5))
    print(coeffs)
    print(r2_score(y_kc, y))

    rmse = sqrt(mean_squared_error(y, y_kc))
    rmse = np.round(rmse, 4)
```

```
    print(rmse)
```

# 7. REFERENCES


Allen, Richard G, Luis S Pereira, Dirk Raes, and Martin Smith. 1998. "Crop Evapotranspiration-Guidelines for Computing Crop Water Requirements-FAO Irrigation and Drainage Paper 56." *Fao, Rome* 300 (9): D05109.

Anderson-Teixeira, Kristina J., John P. Delong, Andrew M. Fox, Daniel A. Brese, and Marcy E. Litvak. 2011. "Differential Responses of Production and Respiration to Temperature and Moisture Drive the Carbon Balance across a Climatic Gradient in New Mexico." *Global Change Biology* 17 (1): 410–424. doi:10.1111/j.1365-2486.2010.02269.x.

Anderson, Martha C, John M Norman, John R Mecikalski, Jason A Otkin, and William P Kustas. 2007. "A Climatological Study of Evapotranspiration and Moisture Stress across the Continental United States Based on Thermal Remote Sensing: 1. Model Formulation." *Journal of Geophysical Research: Atmospheres* 112 (D10). Wiley Online Library.

Baldocchi, Dennis. 2016. *AmeriFlux US-Ton Tonzi Ranch*. AmeriFlux; Univ. of California, Berkeley, CA (United States).

Baldocchi, Dennis D. 2003. "Assessing the Eddy Covariance Technique for Evaluating Carbon Dioxide Exchange Rates of Ecosystems: Past, Present and Future." *Global Change Biology* 9 (4): 479–492. doi:10.1046/j.1365-2486.2003.00629.x.

Barnes, William L., and Vincent V. Salomonson. 2017. "MODIS: A Global Imaging Spectroradiometer for the Earth Observing System." *Optical Technologies for Aerospace Sensing: A Critical Review* 10269 (November 1992): 102690G. doi:10.1117/12.161578.

Berni, J. A.J., P. J. Zarco-Tejada, G. Sepulcre-Cantó, E. Fereres, and F. Villalobos. 2009. "Mapping Canopy Conductance and CWSI in Olive Orchards Using High Resolution Thermal Remote Sensing Imagery." *Remote Sensing of Environment* 113 (11): 2380–2388. doi:10.1016/j.rse.2009.06.018.

Bock, Jane, and Carl Bock. 1986. "The Appleton-Whittell Research Ranch Sanctuary of the National Audubon Society." *CALS Publications Archive. The University of Arizona*.

Bonan, Gordon. 2015. *Ecological Climatology: Concepts and Applications*. Cambridge University Press.

Briske, David D. 2000. *Rangeland Systems*.

Brisson, J., and J. F. Reynolds. 1994. "The Effect of Neighbors on Root Distribution in a Creosotebush (Larrea Tridentata) Population." *Ecology* 75 (6): 1693–1702. doi:10.2307/1939629.

Browne, Jolyon A, Harry E Martz, Stephen G Azevedo, and Joseph W Tringe. 2017. *An Overview of Nondestructive Evaluation and Characterization at Lawrence Livermore National Laboratory*.

Burba, George. 2013. *Eddy Covariance Method for Scientific, Industrial, Agricultural and Regulatory Applications: A Field Book on Measuring Ecosystem Gas Exchange and Areal Emission Rates*. LI-Cor Biosciences.

Bureau of Land Management. 2003. *Santa Rosa and San Jacinto Mountains National Monument Proposed Management Plan and Final Environmental Impact Statement, Riverside County, California. Santa Rosa and San Jacinto Mountains National Monument Final Environmental Impact Statement*. [N. Palm Springs, CA : San Bernardino, CA: U.S. Dept. of the Interior, Bureau of Land Management, Palm Springs-South Coast Field Office ; U.S. Dept. of Agriculture, Forest Service, San Bernardino National Forest, San Jacinto Ranger District. file://catalog.hathitrust.org/Record/101691014.

Campbell, G. S. 1986. "Extinction Coefficients for Radiation in Plant Canopies Calculated Using an Ellipsoidal Inclination Angle Distribution." *Agricultural and Forest Meteorology* 36 (4): 317–321. doi:10.1016/0168-1923(86)90010-9.

Carlsen, Tina M., and Lisa E. Paterson. 2017. "Report on Rare Plant Monitoring and Restoration at the Lawrence Livermore National Laboratory's Experimental Test Site (Site 300) 2015 through 2017" 2081 (18).

Carson, Dee. 2000. "Soils of the Freeman Ranch, Hays County, Texas." *Freeman Ranch Publ. Ser* 4: 1–11.

Chapin, F S. 2002. "III, Matson PA, Mooney H A. Principles of Terrestrial Ecosystem Ecology." New York: Springer-Verlag.

Cleverly, James, James R. Thibault, Stephen B. Teet, Paul Tashjian, Lawrence E. Hipps, Clifford N. Dahm, and Derek Eamus. 2015. "Flooding Regime Impacts on Radiation, Evapotranspiration, and Latent Energy Fluxes over Groundwater-Dependent Riparian Cottonwood and Saltcedar Forests." *Advances in Meteorology* 2015. doi:10.1155/2015/935060.

D'Odorico, Paolo, Jose D. Fuentes, William T. Pockman, Scott L. Collins, Yufei He, Juliana S. Medeiros, Stephan Dewekker, and Marcy E. Litvak. 2010. "Positive Feedback between Microclimate and Shrub Encroachment in the Northern Chihuahuan Desert." *Ecosphere* 1 (6): 1–11. doi:10.1890/ES10-00073.1.

Didan, K. 2015. "MOD13Q1 MODIS/Terra Vegetation Indices 16-Day L3 Global 250m SIN Grid V006." *NASA EOSDIS Land Processes DAAC*.

Doughty, Robin W. 2019. *Unnatural Texas? The Invasive Species Dilemma / Robin W. Doughty and Matt Warnock Turner.* Edited by Matt Warnock Turner. Gideon Lincecum Nature and Environment Series. College Station: Texas A&M University Press.

DuBois, Sean, Ankur R. Desai, Aditya Singh, Shawn P. Serbin, Michael L. Goulden, Dennis D. Baldocchi, Siyan Ma, et al. 2018. "Using Imaging Spectroscopy to Detect Variation in Terrestrial Ecosystem Productivity across a Water-Stressed Landscape." *Ecological Applications* 28 (5): 1313–1324. doi:10.1002/eap.1733.

Dugas, W A, R A Hicks, and Phillip Wright. 1998. "Effect of Removal of Juniperus Ashei on Evapotranspiration and Runoff in the Seco Creek Watershed." *Water Resources Research* 34 (6): 1499–1506.

Emery, Nathan, and Josephine Lesage. 2015. " Late Summer Fog Use In The Drought Deciduous Shrub, Artemisia Californica (Asteraceae) ." *Madroño* 62 (3): 150–157. doi:10.3120/madr-62-03-150-157.1.

Falge, Eva, Dennis Baldocchi, John Tenhunen, Marc Aubinet, Peter Bakwin, Paul Berbigier, Christian Bernhofer, et al. 2002. "Seasonality of Ecosystem Respiration and Gross Primary Production as Derived from FLUXNET Measurements." *Agricultural and Forest Meteorology* 113 (1–4): 53–74. doi:10.1016/S0168-1923(02)00102-8.

Farquhar, G D, and T D Sharkey. 2003. "Stomatal Conductance and Photosynthesis." *Annual Review of Plant Physiology* 33 (1): 317–345. doi:10.1146/annurev.pp.33.060182.001533.

Ferry, R, L Ferry, S Gregory, V Madrid, and J Valett. 2002. "Five-Year Review Report for the Building 834 Operable Unit at Lawrence Livermore National Laboratory Site 300 Environmental Protection Department Environmental Restoration Division," no. February.

Fisher, Joshua B., Dennis D. Baldocchi, Laurent Misson, Todd E. Dawson, and Allen H. Goldstein. 2007. "What the Towers Don't See at Night: Nocturnal Sap Flow in Trees and Shrubs at Two AmeriFlux Sites in California." *Tree Physiology* 27 (4): 597–610. doi:10.1093/treephys/27.4.597.

Friedl, M. A., D. K. McIver, J. C.F. Hodges, X. Y. Zhang, D. Muchoney, A. H. Strahler, C. E. Woodcock, et al. 2002. "Global Land Cover Mapping from MODIS: Algorithms and Early Results." *Remote Sensing of Environment* 83 (1–2): 287–302. doi:10.1016/S0034-4257(02)00078-0.

Gallo, Kevin, Robert Hale, Dan Tarpley, and Yunyue Yu. 2011. "Evaluation of the Relationship between Air and Land Surface Temperature under Clear- and Cloudy-Sky Conditions." *Journal of Applied Meteorology and Climatology* 50 (3): 767–775. doi:10.1175/2010JAMC2460.1.

Gee, G. W., P. J. Wierenga, B. J. Andraski, M. H. Young, M. J. Fayer, and M. L. Rockhold. 1994. "Variations in Water Balance and Recharge Potential at Three Western Desert Sites." *Soil Science Society of America Journal* 58 (1): 63–72. doi:10.2136/sssaj1994.03615995005800010009x.

Goulden, M. L., R. G. Anderson, R. C. Bales, A. E. Kelly, M. Meadows, and G. C. Winston. 2012. "Evapotranspiration along an Elevation Gradient in California's Sierra Nevada." *Journal of Geophysical Research: Biogeosciences* 117 (3): 1–13. doi:10.1029/2012JG002027.

Gurevitch, Jessica, Samuel M Scheiner, and Gordon A Fox. 2002. *The Ecology of Plants*. Sinauer Associates Sunderland.

Gutschick, Vincent P, and Keirith A Snyder. 2006. "Water and Energy Balances within the Jornada Basin." *Structure and Function of a Chihuahuan Desert Ecosystem, The Jornada Basin Long-Term Ecological Research Site. Oxford University Press, New York*, 176–188.

Guyot, Gérard. 1998. *Physics of the Environment and Climate*.

Hastings, Steven J., Walter C. Oechel, and Arturo Muhlia-Melo. 2005. "Diurnal, Seasonal and Annual Variation in the Net Ecosystem CO2 Exchange of a Desert Shrub Community (Sarcocaulescent) in Baja California, Mexico." *Global Change Biology* 11 (6): 927–939. doi:10.1111/j.1365-2486.2005.00951.x.

Hewitt, Ian C., Alexander G. Fernald, and Zohrab A. Samani. 2018. "Calculating Field-Scale Evapotranspiration with Closed-Chamber and Remote Sensing Methods." *Journal of the American Water Resources Association* 54 (4): 962–973. doi:10.1111/1752-1688.12654.

Hu, Leiqiu, Nathaniel A. Brunsell, Andrew J. Monaghan, Michael Barlage, and Olga V. Wilhelmi. 2014. "How Can We Use MODIS Land Surface Temperature to Validate Long-Term Urban Model Simulations?" *Journal of Geophysical Research* 119 (6): 3185–3201. doi:10.1002/2013JD021101.

Huete, A. R., H. Q. Liu, K. Batchily, and W. Van Leeuwen. 1997. "A Comparison of Vegetation Indices over a Global Set of TM Images for EOS-MODIS." *Remote Sensing of Environment* 59 (3): 440–451. doi:10.1016/S0034-4257(96)00112-5.

Huete, A R, K Didan, Alfredo Huete, Kamel Didan, Wim Van Leeuwen, Andrea Jacobson, Ramon Solanos, and Trevor Laing. 1999. "MODIS VEGETATION INDEX (MOD 13) ALGORITHM THEORETICAL BASIS DOCUMENT Principal Investigators Development Team MODIS Product ID: MOD13," no. April 1999. http://vip.arizona.edu.

Huxman, Travis E., Bradford P. Wilcox, David D. Breshears, Russell L. Scott, Keirith A. Snyder, Eric E. Small, Kevin Hultine, William T. Pockman, and Robert B. Jackson. 2005. "Ecohydrological Implications of Woody Plant Encroachment." *Ecology* 86 (2): 308–319. doi:10.1890/03-0583.

Jaksic, Vesna, Gerard Kiely, John Albertson, Ram Oren, Gabriel Katul, Paul Leahy, and Kenneth A. Byrne. 2006. "Net Ecosystem Exchange of Grassland in Contrasting Wet and Dry Years." *Agricultural and Forest Meteorology* 139 (3–4): 323–334. doi:10.1016/j.agrformet.2006.07.009.

Jimenez-Munoz, Juan C., Jose A. Sobrino, Drazen Skokovic, Cristian Mattar, and Jordi Cristobal. 2014. "Land Surface Temperature Retrieval Methods from Landsat-8 Thermal Infrared Sensor Data." *IEEE Geoscience and Remote Sensing Letters* 11 (10): 1840–1843. doi:10.1109/LGRS.2014.2312032.

Kamble, Baburao, Ayse Kilic, and Kenneth Hubbard. 2013. "Estimating Crop Coefficients Using Remote Sensing-Based Vegetation Index." *Remote Sensing* 5 (4): 1588–1602. doi:10.3390/rs5041588.

Kellogg, Elizabeth A. 2013. "C4 Photosynthesis." *Current Biology* 23 (14): R594–R599. doi:https://doi.org/10.1016/j.cub.2013.04.066.

Kimball, Sarah, Megan E. Lulow, Kailen A. Mooney, and Quinn M. Sorenson. 2014. "Establishment and Management of Native Functional Groups in Restoration." *Restoration Ecology* 22 (1): 81–88. doi:10.1111/rec.12022.

Kjelgaard, J. F., J. L. Heilman, K. J. McInnes, M. K. Owens, and R. H. Kamps. 2008. "Carbon Dioxide Exchange in a Subtropical, Mixed C3/C4 Grassland on the Edwards Plateau, Texas." *Agricultural and Forest Meteorology* 148 (6–7): 953–963. doi:10.1016/j.agrformet.2008.01.006.

Ko, Jonghan, and Giovanni Piccinni. 2009. "Corn Yield Responses under Crop Evapotranspiration-Based Irrigation Management." *Agricultural Water Management* 96 (5): 799–808. doi:10.1016/j.agwat.2008.10.010.

Kool, D., N. Agam, N. Lazarovitch, J. L. Heitman, T. J. Sauer, and A. Ben-Gal. 2014. "A Review of Approaches for Evapotranspiration Partitioning." *Agricultural and Forest Meteorology* 184. Elsevier B.V.: 56–70. doi:10.1016/j.agrformet.2013.09.003.

Krishnan, Praveena, Tilden P. Meyers, Russell L. Scott, Linda Kennedy, and Mark Heuer. 2012. "Energy Exchange and Evapotranspiration over Two Temperate Semi-Arid Grasslands in North America." *Agricultural and Forest Meteorology* 153. Elsevier B.V.: 31–44. doi:10.1016/j.agrformet.2011.09.017.

Lambers, Hans, F Stuart Chapin III, and Thijs L Pons. 2008. *Plant Physiological Ecology*. Springer Science & Business Media.

Litvak, Marcy. 2016a. *AmeriFlux US-Wjs Willard Juniper Savannah*. AmeriFlux; University of New Mexico.

Litvak, Marcy. 2016b. *AmeriFlux US-Mpj Mountainair Pinyon-Juniper Woodland*. AmeriFlux; University of New Mexico.

Litvak, Marcy. 2016c. *AmeriFlux US-FR2 Freeman Ranch-Mesquite Juniper*. AmeriFlux; University of New Mexico.

Ma, Siyan, Dennis Baldocchi, Sebastian Wolf, and Joseph Verfaillie. 2016. "Slow Ecosystem Responses Conditionally Regulate Annual Carbon Balance over 15 Years in Californian Oak-Grass Savanna." *Agricultural and Forest Meteorology* 228–229. Elsevier B.V.: 252–264. doi:10.1016/j.agrformet.2016.07.016.

Martínez-Cob, A. 2008. "Use of Thermal Units to Estimate Corn Crop Coefficients under Semiarid Climatic Conditions." *Irrigation Science* 26 (4): 335–345. doi:10.1007/s00271-007-0097-5.

McClaran, M. P., and D. L. Angell. 2006. "Long-Term Vegetation Response to Mesquite Removal in Desert Grassland." *Journal of Arid Environments* 66 (4): 686–697. doi:10.1016/j.jaridenv.2005.12.010.

McClaran, Mitchel P. 2003. "A Century of Vegetation Change on the Santa Rita Experimental Range." *Santa Rita Experimental Range* 100: 16–33.

McCole, Amy A., and Libby A. Stern. 2007. "Seasonal Water Use Patterns of Juniperus Ashei on the Edwards Plateau, Texas, Based on Stable Isotopes in Water." *Journal of Hydrology* 342 (3–4): 238–248. doi:10.1016/j.jhydrol.2007.05.024.

Meyer, T., and G. S. Okin. 2015. "Evaluation of Spectral Unmixing Techniques Using MODIS in a Structurally Complex Savanna Environment for Retrieval of Green Vegetation, Nonphotosynthetic Vegetation, and Soil Fractional Cover." *Remote Sensing of Environment* 161. Elsevier Inc.: 122–130. doi:10.1016/j.rse.2015.02.013.

Mustard, John F. 2017. "From Planets to Crops and Back: Remote Sensing Makes Sense." *Journal of Geophysical Research: Planets* 122 (4): 794–797. doi:10.1002/2017JE005315.

Nagler, Pamela L., Edward P. Glenn, T. Lewis Thompson, and Alfredo Huete. 2004. "Leaf Area Index and Normalized Difference Vegetation Index as Predictors of Canopy Characteristics and Light Interception by Riparian Species on the Lower Colorado River." *Agricultural and Forest Meteorology* 125 (1–2): 1–17. doi:10.1016/j.agrformet.2004.03.008.

Nagler, Pamela L., Edward P. Glenn, Uyen Nguyen, Russell L. Scott, and Tanya Doody.

2013. “Estimating Riparian and Agricultural Actual Evapotranspiration by Reference Evapotranspiration and MODIS Enhanced Vegetation Index.” *Remote Sensing* 5 (8): 3849–3871. doi:10.3390/rs5083849.
Neuenschwander, Amy L. 2009. “Landcover Classification of Small-Footprint, Full-Waveform Lidar Data.” *Journal of Applied Remote Sensing* 3 (1): 033544. doi:10.1117/1.3229944.
Norman, John M, William P Kustas, and Karen S Humes. 1995. “Source Approach for Estimating Soil and Vegetation Energy Fluxes in Observations of Directional Radiometric Surface Temperature.” *Agricultural and Forest Meteorology* 77 (3–4). Elsevier: 263–293.
Ochoa, Carlos G., Alexander G. Fernald, Steven J. Guldan, Vincent C. Tidwell, and Manoj K. Shukla. 2013. “Shallow Aquifer Recharge from Irrigation in a Semiarid Agricultural Valley in New Mexico.” *Journal of Hydrologic Engineering* 18 (10): 1219–1230. doi:10.1061/(ASCE)HE.1943-5584.0000718.
Okin, Gregory S., Dar A. Roberts, Bruce Murray, and William J. Okin. 2001. “Practical Limits on Hyperspectral Vegetation Discrimination in Arid and Semiarid Environments.” *Remote Sensing of Environment* 77 (2): 212–225. doi:10.1016/S0034-4257(01)00207-3.
Paruelo, Jose M, and W K Lauenroth. 1996. “Relative Abundance of Plant Functional Types in Grasslands and Shrublands of North America.” *Ecological Applications* 6 (4). Wiley Online Library: 1212–1224.
Pontailler, Jean Yves, Graham J. Hymus, and Bert G. Drake. 2003. “Estimation of Leaf Area Index Using Ground-Based Remote Sensed NDVI Measurements: Validation and Comparison with Two Indirect Techniques.” *Canadian Journal of Remote Sensing* 29 (3): 381–387. doi:10.5589/m03-009.
Potter, Christopher. 2018. “Shifts of Vegetation Cover on Southern California Mountains in Response to Recent Climate Variations.” *Journal of Earth and Environmental Sciences* 2018 (01): 1–9. doi:https://doi.org/10.1007/s41976-019-00013-x.
Potts, D. L., K. N. Suding, G. C. Winston, A. V. Rocha, and M. L. Goulden. 2012. “Ecological Effects of Experimental Drought and Prescribed Fire in a Southern California Coastal Grassland.” *Journal of Arid Environments* 81. Elsevier Ltd: 59–66. doi:10.1016/j.jaridenv.2012.01.007.
Preston, R E. 2006. “Special-Status Plant Species Surveys and Vegetation Mapping at Lawrence Livermore National Laboratory.”
Ratajczak, Zakary, Jesse B. Nippert, and Scott L. Collins. 2012. “Woody Encroachment Decreases Diversity across North Americangrasslands and Savannas.” *Ecology* 93 (4): 697–703.
Rawls, W.J., D.L. Brakensiek, and K.E. Saxton. 1982. “Rawls et Al 1982.Pdf.” *Transactions of the ASAE*. http://www.envsci.rutgers.edu/~gimenez/SoilPhysics/Homework08/Rawls et al 1982.pdf.
Renard, K. G., M. H. Nichols, D. A. Woolhiser, and H. B. Osborn. 2008. “A Brief Background on the U.S. Department of Agriculture Agricultural Research Service

Walnut Gulch Experimental Watershed.” *Water Resources Research* 44 (5): 1–11. doi:10.1029/2006WR005691.
Sage, Rowan F, David A Wedin, and Meirong Li. 1999. “The Biogeography of C4 Photosynthesis: Patterns and Controlling Factors.” *C4 Plant Biology*. Academic Press San Diego, 313–373.
Samani, Zohrab, A. Salim Bawazir, Max Bleiweiss, Rhonda Skaggs, John Longworth, Vien D. Tran, and Aldo Pinon. 2009. “Using Remote Sensing to Evaluate the Spatial Variability of Evapotranspiration and Crop Coefficient in the Lower Rio Grande Valley, New Mexico.” *Irrigation Science* 28 (1): 93–100. doi:10.1007/s00271-009-0178-8.
Sayre, Nathan F. 2017. *The Politics of Scale: A History of Rangeland Science*. University of Chicago Press.
Schreiner-McGraw, Adam P., and Enrique R. Vivoni. 2017. “Percolation Observations in an Arid Piedmont Watershed and Linkages to Historical Conditions in the Chihuahuan Desert.” *Ecosphere* 8 (11). doi:10.1002/ecs2.2000.
Scott, Russell L., Joel A. Biederman, Erik P. Hamerlynck, and Greg A. Barron-Gafford. 2015. “The Carbon Balance Pivot Point of Southwestern U.S. Semiarid Ecosystems: Insights from the 21st Century Drought.” *Journal of Geophysical Research: Biogeosciences* 120 (12): 2612–2624. doi:10.1002/2015JG003181.
Scott, Russell L., Travis E. Huxman, William L. Cable, and William E. Emmerich. 2006. “Partitioning of Evapotranspiration and Its Relation to Carbon Dioxide Exchange in a Chihuahuan Desert Shrubland.” *Hydrological Processes: An International Journal* 20 (15): 3227–3243. doi:10.1002/hyp.
Scott, Russell L., G. Darrel Jenerette, Daniel L. Potts, and Travis E. Huxman. 2009. “Effects of Seasonal Drought on Net Carbon Dioxide Exchange from a Woody-Plant-Encroached Semiarid Grassland.” *Journal of Geophysical Research: Biogeosciences* 114 (4): 1–13. doi:10.1029/2008JG000900.
Senay, Gabriel B, Stefanie Bohms, Ramesh K Singh, Prasanna H Gowda, Naga M Velpuri, Henok Alemu, and James P Verdin. 2013. “Operational Evapotranspiration Mapping Using Remote Sensing and Weather Datasets: A New Parameterization for the SSEB Approach.” *JAWRA Journal of the American Water Resources Association* 49 (3). Wiley Online Library: 577–591.
Sher, Anna, A Sher, Martin F Quigley, and M F Quigley. 2013. *Tamarix: A Case Study of Ecological Change in the American West*. Oxford University Press. doi:10.1093/acprof:osobl/9780199898206.001.0001.
Stofan, E. R., C. Elachi, J. I. Lunine, R. D. Lorenz, B. Stiles, K. L. Mitchell, S. Ostro, et al. 2007. “The Lakes of Titan.” *Nature* 445 (7123): 61–64. doi:10.1038/nature05438.
Tempfli, Klaus, GCe Huurneman, WHe Bakker, Lucas L F Janssen, W F Feringa, A S M Gieske, K A Grabmaier, C A Hecker, J A Horn, and N Kerle. 2009. “Principles of Remote Sensing: An Introductory Textbook.” In . ITC.
Timmermans, Wim J., William P. Kustas, Martha C. Anderson, and Andrew N. French. 2007. “An Intercomparison of the Surface Energy Balance Algorithm for Land

(SEBAL) and the Two-Source Energy Balance (TSEB) Modeling Schemes." *Remote Sensing of Environment* 108 (4): 369–384. doi:10.1016/j.rse.2006.11.028.
Vinukollu, Raghuveer K., Eric F. Wood, Craig R. Ferguson, and Joshua B. Fisher. 2011. "Global Estimates of Evapotranspiration for Climate Studies Using Multi-Sensor Remote Sensing Data: Evaluation of Three Process-Based Approaches." *Remote Sensing of Environment* 115 (3). Elsevier Inc.: 801–823. doi:10.1016/j.rse.2010.11.006.
Wan, Z, S Hook, and G Hulley. 2015. "MOD11A2 MODIS/Terra Land Surface Temperature/Emissivity 8-Day L3 Global 1km SIN Grid V006." *NASA EOSDIS Land Processes DAAC* 10.
Wang, Chuanli, Longyun Guo, Yixue Li, and Zhuo Wang. 2012. "Systematic Comparison of C3 and C4 Plants Based on Metabolic Network Analysis." *BMC Systems Biology* 6 (SUPPL.2). doi:10.1186/1752-0509-6-S2-S9.
Wang, Jie, Xiangming Xiao, Yao Zhang, Yuanwei Qin, Russell B. Doughty, Xiaocui Wu, Rajen Bajgain, and Ling Du. 2018. "Enhanced Gross Primary Production and Evapotranspiration in Juniper-Encroached Grasslands." *Global Change Biology* 24 (12): 5655–5667. doi:10.1111/gcb.14441.
Weiss, Jeremy L., David S. Gutzler, Julia E.Allred Coonrod, and Clifford N. Dahm. 2004. "Long-Term Vegetation Monitoring with NDVI in a Diverse Semi-Arid Setting, Central New Mexico, USA." *Journal of Arid Environments* 58 (2): 249–272. doi:10.1016/j.jaridenv.2003.07.001.
Wharton, Sonia, Jennifer Newman, Jessica Osuna, Siyan Ma, Matthias Falk, Ken Bible, and Dennis Baldocchi. 2013. "Atmospheric LIDAR Provides Insight into Land Surface-Atmosphere Exchange at 3 AmeriFlux Towers."
Whitford, Walter G. 1997. "Desertification and Animal Biodiversity in the Desert Grasslands of North America." *Journal of Arid Environments* 37: 709–720.
Williams, D. G., W. Cable, K. Hultine, J. C.B. Hoedjes, E. A. Yepez, V. Simonneaux, S. Er-Raki, et al. 2004. "Evapotranspiration Components Determined by Stable Isotope, Sap Flow and Eddy Covariance Techniques." *Agricultural and Forest Meteorology* 125 (3–4): 241–258. doi:10.1016/j.agrformet.2004.04.008.
Williamson, Scott N., David S. Hik, John A. Gamon, Jeffrey L. Kavanaugh, and Saewan Koh. 2013. "Evaluating Cloud Contamination in Clear-Sky MODIS Terra Daytime Land Surface Temperatures Using Ground-Based Meteorology Station Observations." *Journal of Climate* 26 (5): 1551–1560. doi:10.1175/JCLI-D-12-00250.1.
Wu, Taixia, Lifu Zhang, and Changping Huang. 2015. "An Analysis of Shadow Effects on Spectral Vegetation Indices Using a Ground-Based Imaging Spectrometer." *Workshop on Hyperspectral Image and Signal Processing, Evolution in Remote Sensing* 2015-June (11): 2188–2192. doi:10.1109/WHISPERS.2015.8075503.
Yousif, Charles, Gorka Oña Quecedo, and Julia Bilbao Santos. 2013. "Comparison of Solar Radiation in Marsaxlokk, Malta and Valladolid, Spain." *Renewable Energy* 49. Elsevier Ltd: 203–206. doi:10.1016/j.renene.2012.01.031.
Zotarelli, Lincoln, and Md Dukes. 2010. "Step by Step Calculation of the Penman-

Monteith Evapotranspiration (FAO-56 Method)." *Institute of Food and …*, 1–10. doi:10.1016/j.scitotenv.2011.10.067.